901画像処理ライブラリ
FVL基本統合SDK/902(FVL-SDK-ITG/902)
903画像処理ライブラリ
FVL基本SDK/904(FVL-SDK-BSC/904)
FVL基本SDK/LNX(FVL-SDK-BSC/LNX)

90X
基本ライブラリ説明書
～画像入力編～

Vol.1

☆第25版☆

(株)ファースト

はしがき

この説明書は、

901画像処理ライブラリ　Ver.6.50

FVL基本統合SDK/902(FVL-SDK-ITG/902) Ver.5.10

903画像処理ライブラリ　Ver.2.00

FVL基本SDK/904(FVL-SDK-BSC/904) Ver.2.00

FVL基本SDK/LNX Ver.3.10

に対応しています。

本書は、90Xシリーズ用ユーザプログラム作成時に使用できる、「基本ライブラリ」について記載したものです。

なお、この他に90Xシリーズライブラリとしては、下記のようなものがあります。

◎90X　基本ライブラリ(Vol.2, Vol.3)説明書

◎90X　2値画像ライブラリ説明書

◎90X　濃淡画像ライブラリ説明書

◎90X　ビジョン・ツール・ライブラリ説明書

◎90X　キャリパーライブラリ説明書

◎902　高分解能センサライブラリ説明書

ユーザプログラムの開発方法その他につきましては

◎FAST Vision Libraryプログラマーズガイド

◎90X　操作説明書

◎FVL/LNX　操作説明書

をご参照下さい。

[注1]901用のコンパイラ MCP960(960 Cクロスコンパイラ)のバージョンは 2.0以上です。

[注2]902用のコンパイラ Watcom Cのバージョンは10.5以上です。

[注3]903, 904用のコンパイラ Green Hills C および MULTIのバージョンは1.8.9 以上です。

この説明書で、

RICE-001と記載されている場合、RICE-001、RICE-001a、RICE-001b

FVC04と記載されている場合、FVC04、FVC04a

FV904と記載されている場合、FV904、FV904a、FV904ac、FV904p、FV904pc

を指します。

目　　次

目　　次

1. 概　　要

本書は基本機能をライブラリの形で供給するほかに、90X シリーズの保有するハードウェアの制御サービス、その他を供給するものです。
本書(Vol.2)では、各ライブラリについて、下記のビデオ入力制御ライブラリについて記載します。

Vol. 1

1. ビデオ入力制御ライブラリ

Vol. 2

1. グラフィックライブラリ
2. メモリ領域割当ライブラリ
3. メモリ操作ライブラリ
4. 画像計測ライブラリ
5. キー操作ライブラリ
6. 標準入出力ライブラリ
7. RS232Cライブラリ
8. C-NET基本ライブラリ
9. C-NET応用ライブラリ
10. C-LINKライブラリ
11. パラレル・リンクライブラリ
12. E-LINKライブラリ
13. 外部入出力端子ライブラリ
14. 時刻サービスライブラリ
15. 数値演算ライブラリ
16. 特殊計測・計算処理ライブラリ
17. アフィン変換ライブラリ
18. 座標変換（キャリブレーション）ライブラリ
19. カーソル関連ライブラリ

Vol. 3

1. パッド関連ライブラリ
2. メニュー・バー関連ライブラリ
3. リスト（表形式）関連ライブラリ
4. 照明制御ライブラリ
5. ファイルシステムライブラリ
6. システム制御ライブラリ
7. システムパラメータ関連ライブラリ
8. フォトカプラ絶縁型I/O拡張ライブラリ

なお、各ライブラリ関数の表記

901	902	903	904	FVL/LNX	高分解能

は、901, 902, 903, 904, FVL/LNX及び高分解能カメラ使用時にライブラリ関数が使用できるかどうかを表すもので

○ ············ 使用可能
△ ············ 一部使用可能
空白············ 使用不可

となっております。
△の場合は、必ず留意事項を参照してください。

2. ビデオ入力制御ライブラリ

ビデオ制御ライブラリは、製品により使用できる機能が異なります。
お客様が購入された製品が、ビデオ制御ライブラリのどの機能が利用可能かは留意事項を参照してください。
留意事項では、FV2X00，CSC901，CSC903，FV904シリーズの場合、製品名により使用可能な機能を知ることができます。
CSC902，FV902，FV2X00シリーズ，FV2X10シリーズ，FV3X00シリーズは、同じ製品名でも搭載されている映像キャプチャボード(映像入力ボード)により機能が異なりますので、購入されたCSC902，FV902，FV2X00シリーズ，FV3X00シリーズに、どの映像キャプチャボードが搭載されているか知ることが必要です。CSC902，FV902，FV2X00シリーズ，FV3X00シリーズに、どのキャプチャボードが搭載されているかはシステム情報のキャプチャボードを参照してください。この機能は、システム・バージョン2.20から使用可能です。
詳細は90X操作説明書「付録・電源投入時の特殊操作」を参照してください。

CSC902，FV902，FV2X00，FV2X10，FV3X00搭載キャプチャボード

キャプチャ	読み方	搭載・製品名
RICE	ライス	CSC902ST－RICE FV902 FV2X00-DOS FV2X00-LNX FV3X00-DOS
FVC01	エフブイシーゼロイチ	FV902 FV2X00-DOS FV3X00-DOS
FVC02	エフブイシーゼロニ	FV902 FV2X00-DOS FV2X00-LNX FV3X00-DOS
FVC04	エフブイシーゼロヨン	FV2X00-LNX
FVC05	エフブイシーゼロゴ	FV2X00-DOS FV2X00-LNX FV3X00-DOS
FVC06	エフブイシーゼロロク	FV2X00-LNX
FV-GP440	エフブイジーピー ヨンヨンマル	FV2X10-LNX
FHC3312	エフエッチシー サンサンイチニ	FV2X10-LNX
FHC3313	エフエッチシー サンサンイチサン	FV2X10-LNX

ビデオ制御ライブラリとしては、下記が実施できます。

■メモリビデオ入力

入力ビデオにより画像メモリの内容を逐次更新(＝フリーラン)するか、現在の状況を凍結(＝フリーズ)するかを指令するものです。

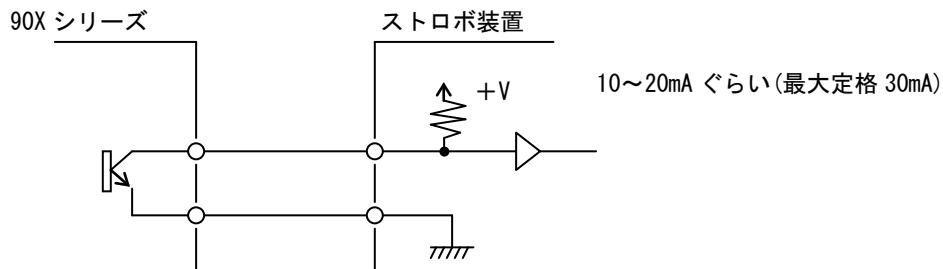
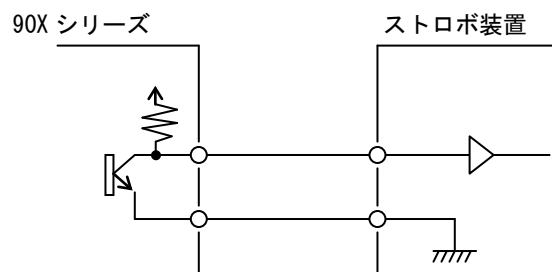
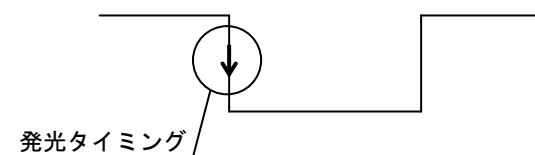
■ストロボ制御

TVカメラが映像をキャッチする際の光源制御を意味し、動いているものの瞬間の姿を撮し取る場合に用いられます。

ストロボ装置とのインターフェイスについて

1. 90Xシリーズには、ストロボ発光のタイミング信号の出力があるものがあります。

2. 出力形式はオープンコレクタとプルアップ出力のいずれかをソフトウエア
   (Lib_set_strobo)により選択できますので、ストロボ装置の仕様に合わせてください。

   (1)オープンコレクタ出力の場合

   (2)プルアップ出力の場合

3. 発光タイミングは、出力メークのタイミング(パルスの立下がりエッジ)で取ってください。
   (ストロボ装置の仕様と合致している必要があります。)

■フレームバッファ転送

画像メモリ上の内容をモニタTV上に表示するためのものです。

■2値入力/濃淡入力の選択

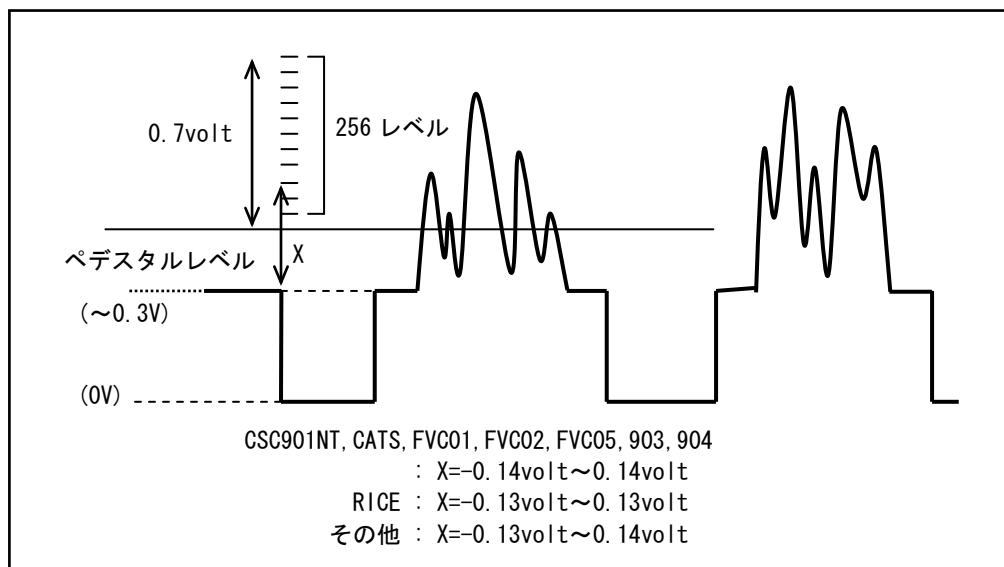
90Xシリーズの画像イメージデータを格納しておくための領域は、メインメモリの一部分が割り当てられます。
濃淡画像の場合は、8ビット/1画素で、512×480 の濃淡画像メモリを1面から任意の面数を保有できます。
また、2値画像の場合は、1ビット/1画素(ビット・マップ)で、512×480 の2値画像メモリを1面から任意の面数を保有できます。
この2種の画像メモリのどちら側、または2種共にビデオ入力画像を入力するか選択する必要があります。

■ビデオ入力チャネル指定･切換制御

90Xシリーズには4又は2つのビデオ入力チャネルが用意されており、選択された1チャネル(1部機種では、複数チャネル) からのビデオ入力ができるようになっております。後述の各種制御のうち、下記のものについては、チャネル毎に独立可能です。

☆ビデオ入力オフセット
☆ビデオ2値化制御

■ビデオ入力オフセット

TVカメラから送信されるアナログ信号(1volt peak to peak)のうちのペデスタルレベル(約 0.3v)より上の0.7vの範囲を256等分して濃度値を得ています。
この意味では、0はペデスタルレベルの電圧値、255はペデスタルレベル+0.7vの電圧値に相当するわけですが、このオフセット制御により”ペデスタルレベル-X “voltを0とする事ができ、かつ、Xとしては下記のようにプログラム的に変化させる事ができます。

■ビデオ2値化制御

256段階に量子化された入力ビデオ白(=1), 黒(=0)の2値に変換する際の閾値を与えるものです。

2.1　ストロボ入力のサンプルプログラム

①901, 904 用

```
#include “f_gui.h”
#include ″f_video.h″
#include ″f_time.h″

void ajust( void );             /* カメラのセッティングのための　連続発行入力 */
void set( void );               /* ストロボフリーズのための初期設定 */
void run( void );               /* ストロボフリーズ */



void main( void )
{
     Lib_init_cursor();/* カーソル操作のための初期化 */

     Lib_display_keyinput( 10, 450, ″連続入力″ );
     ajust();                   /* カメラのセッティングのための　連続発行入力 */

     set();                     /* ストロボフリーズのための　初期設定 */
     while ( 2 != Lib_display_keyinput( 10, 450, ″フリーズ″ ) )
     {
         run();                 /* ストロボフリーズ */
     }
}


void ajust( void )              /* カメラのセッティングのための　連続発行入力 */
{
     Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );       /* 設定 */
     Lib_ctrl_strobo( START_STROBO );                 /* ストロボ動作開始 */
     Lib_freerun();                                   /* 連続入力 */
     Lib_display_keyinput( 10, 450, ″停止″ );         /* クリック入力で停止 */
     Lib_ctrl_strobo( STOP_STROBO );                  /* ストロボ動作停止 */
}


void set( void )                /* ストロボフリーズのための初期設定 */
{
     Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );       /* 設定 */
     Lib_time_delay( 33 );                      /* 設定が確定するまで時間待ち */
}


void run( void )                /* ストロボフリーズ */
{
     Lib_strobo_freeze( TRANSMIT );
}
```

※FV904p、FV904pc の場合はストロボ端子が無い為、上記サンプルプログラムは使用できません。

②902 用

```
#include ″f_gui.h″
#include ″f_video.h″
#include ″f_time.h″

void ajust( void );             /* カメラのセッティングのための　連続発行入力 */
void run( void );               /* ストロボフリーズ */



void main( void )
{
     Lib_init_cursor();         /* カーソル操作のための初期化 */

     Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );      /* 設定 */
     Lib_set_video_input_mode( STROBO_VIDEO_MODE, 0 );

     Lib_display_keyinput( 10, 450, ″連続入力″ );
     ajust();                   /* カメラのセッティングのための　連続発行入力 */

     while ( 2 != Lib_display_keyinput( 10, 450, ″フリーズ″ ) )
     {
         run();                 /* ストロボフリーズ */
     }
}


void ajust( void )              /* カメラのセッティングのための　連続発行入力 */
{
     int  x, y;

     while ( 1 )                /* キィが押されるまで　連続発行入力 */
     {
         Lib_freeze( TRANSMIT );
         if ( 0 != Lib_see_current_position( &x, &y ) )
             break;
     }
}


void run( void )                /* ストロボフリーズ */
{
     Lib_freeze( TRANSMIT );
}
```

2.2　903について

■接続カメラ

チャネル0と1には、必ず同じ種類のカメラを接続してください。

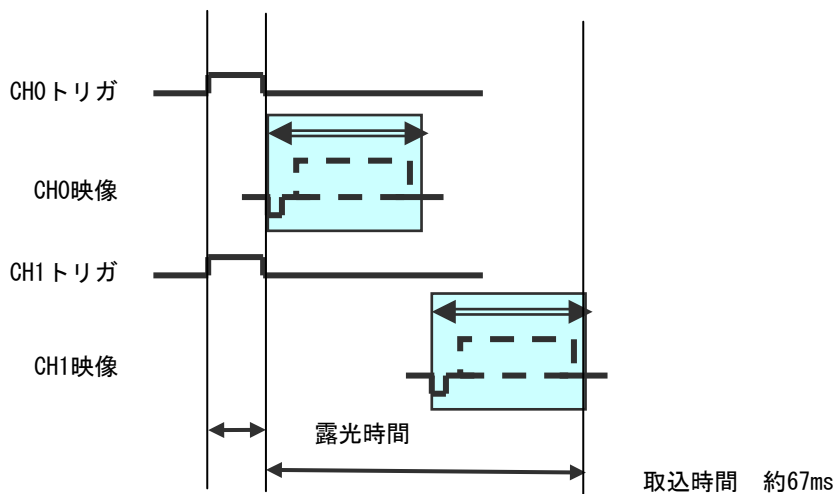
2008.04.01現在のサポートカメラは以下のようになります。

カメラ	メーカー	補足
XC-55	ソニー製	30F 1/3
CS8530-01	東芝テリー製	30F 1/3
CS8550i-01	東芝テリー製(※)	30F 1/3
CS8530D-01	東芝テリー製	60F 1/3
CS8550Di-01	東芝テリー製(※)	60F 1/3

(※)画像垂直取込ライン設定関数で+12指定することで、CS8530-01と同等の取込が可能です。

■903の同時露光

CSC903は同時露光をサポートしています。

同時露光では、チャネル0とチャネル1の両方で同時に露光が行われ、映像は、指定された優先順位に従い、順番に両チャネルとも画像メモリへ転送されます。

・関連ライブラリ

Lib_freeze
Lib_set_exposure
Lib_random_bin_level
Lib_ena_ext_trg
Lib_dis_ext_trg

2.3　904について

FV904シリーズは、FV904、FV904a、FV904ac、FV904p、FV904pcの5機種あります。

■接続カメラ
機種ごとに接続可能なカメラが異なります。最大4台(CH0～CH3)接続出来ます。
CH0～CH3には必ず同じ種類のカメラを接続して下さい。

FV904、FV904a、FV904acでサポートされるカメラ

EIA170方式（RS-170）に準拠した白黒ビデオカメラ

FV904p、FV904pcでサポートされるカメラ

2008.04.01 現在のサポートカメラは以下のようになります。

カメラ	メーカー	補足
XC-55/XC56	ソニー製	30F 1/3（XC-56互換カメラ）
CS8550i-01	東芝テリー製(※)	30F 1/3（XC-56互換カメラ）
XC-HR50/XC-HR57	ソニー製	60F HR50:1/3, HR57:1/2
CS8550D/CS8550Di	東芝テリー製(※)	60F 1/3
CS8560D/CS8570D	東芝テリー製	60F CS8560D:1/3 CS8570D:1/2
KP-F30	日立国際電気製(※)	60F 1/3
KP-F33	日立国際電気製(※)	30F 1/3
IK-52V/53V	東芝製	60F IK-52V:1/2 IK-53V:1/3

(※)FV904p、FV904pcではXC-55/56で、ほぼカメラの中心部分が取り込まれる様にタイミングを設定しています。よってXC-55/56とタイミングの異なるカメラは、画像垂直取込ライン設定関数で調整が必要です。

XC-55/56とタイミングの異なるカメラ
CS8550i-01, CS8550D/CS8550Di　・・・・・ +12
KP-F30,KP-F33　・・・・・ -6

KP-F30は同時に接続可能な台数は3台までとなります。

XC-56互換カメラを使用する場合とそれ以外のカメラを使用する場合で904システム上で「トリガ出力ピン」の設定を変更する必要があります。初期値は「Disable」(無接続)となっていますので使用するカメラに合わせて最初に必ず1度は設定して下さい。設定後システムパラメータの保存(システムのメイン画面より○システム＞○セーブ)を実行すれば次回起動時より設定が有効となります。

「トリガ出力ピン」設定方法について

FV904p, FV904pc使用時「トリガ出力ピン」を切り替えるには、システムメニューより
〇システム ＞ 〇オプション ＞ 〇システム構築 ＞〇トリガ出力ピンを選択します。

FV904p使用時システムメニュー
「トリガ出力ピンの設定」画面

■露光期間出力
FV904p, FV904pcではランダムトリガで取込を行う場合、露光期間中に、外部機器制御コネクタ(EXT.CONTROL)より露光期間信号が出力されます。(露光期間出力設定が「露光期間出力許可」になっている場合(※))

(※)露光期間信号を出力するには、露光期間出力設定関数で「露光期間出力許可」にする必要があります。ただし初期設定は「露光期間出力許可」となっていますので装置起動後1度も変更していない場合は何も設定しなくても露光期間信号は出力されます。

外部機器制御コネクタ(EXT.CONTROL)を汎用入出力として使用したい場合は、露光期間出力設定関数で「露光期間出力禁止」にする必要があります。露光期間出力との併用は出来ません。

2.4　FVC01について

■サポートカメラ
2004.04.01 現在のサポートカメラは以下のようになります。

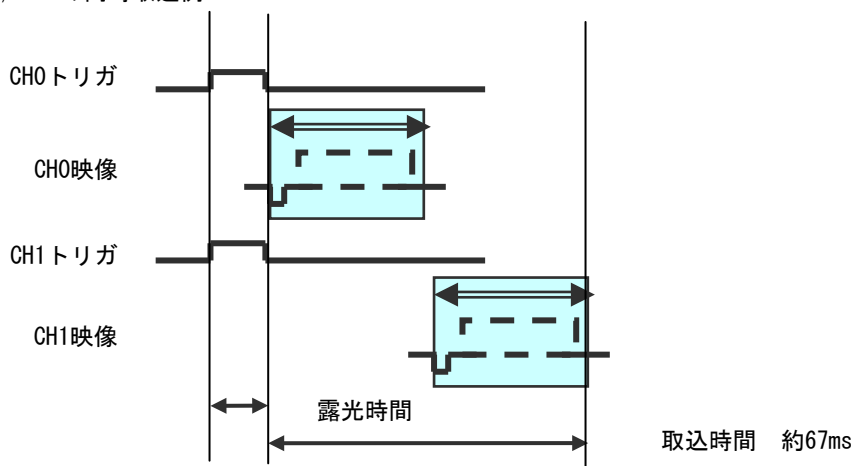
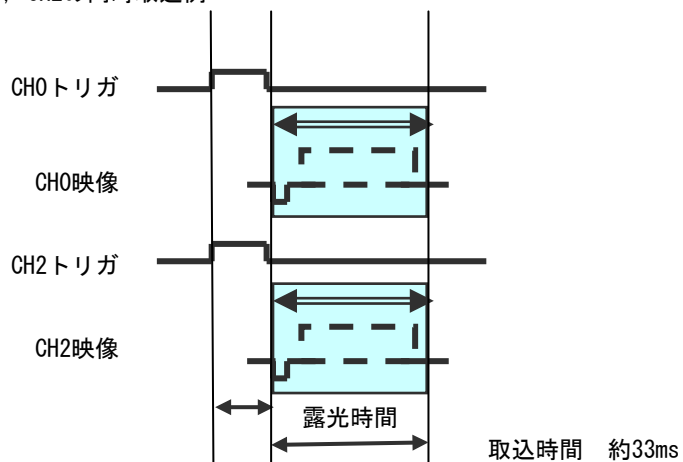
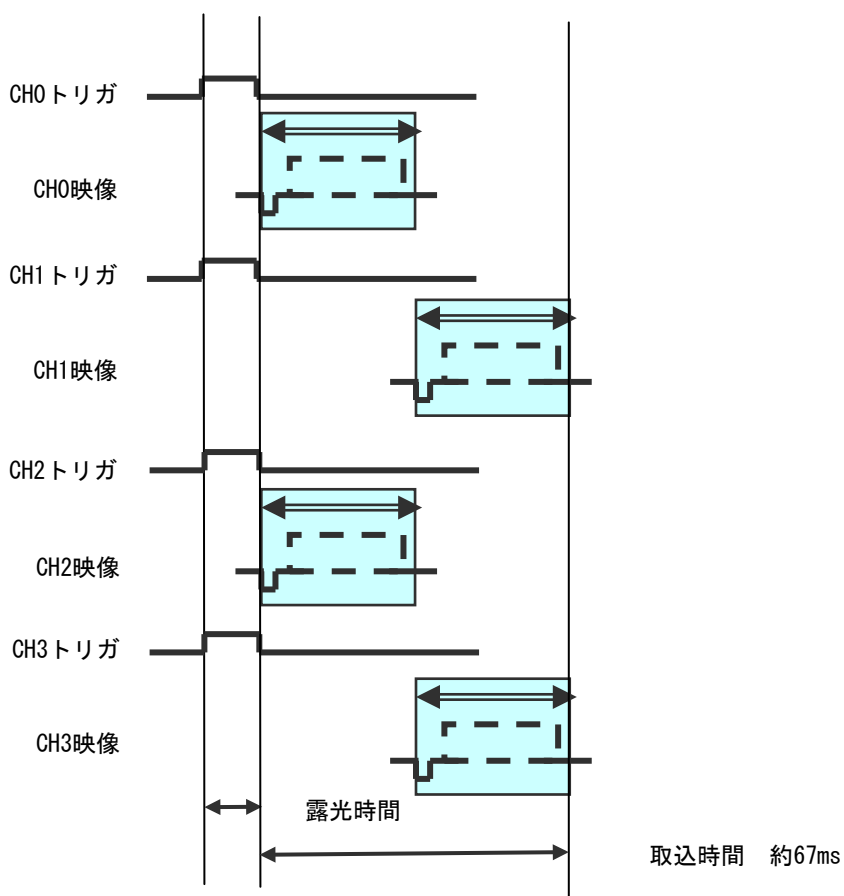
カメラ	メーカー	補足
XC-55	ソニー製	30F 1/3
CS8530-01	東芝テリー製	30F 1/3
CS8550i-01	東芝テリー製(※)	30F 1/3
CS8530D-01	東芝テリー製	60F 1/3
CS8550Di-01	東芝テリー製(※)	60F 1/3

(※)画像垂直取込ライン設定関数で+12指定することで、CS8530-01と同等の取込が可能です。

■接続カメラ
チャネル0と1には、必ず同じ種類のカメラを接続してください。

■2枚搭載時のチャネル割付
FVC01が2枚搭載されている場合のチャネルは、以下の表になります。

搭載番号	チャネル番号
1枚目	CH0, 1
2枚目	CH2, 3

■同時取込
FVC01は、2枚実装時に同時取込をサポートします(1枚実装時は、同時露光となります)。
同時取込を行うチャネル指定が、1枚のボード内(例えばCH0, CH1)の場合は、同時露光として動作し、2枚にまたがる(例えばCH0, CH2)場合は、同時取込として動作します。
外部ランダムトリガの場合、同じボード内の取込要求 TRGRQ0、TRGRQ1は同期連動します。
いずれか1つの信号を入力してください。異なるボード間の取込要求は同期連動しません。必ずそれぞれのボードの取込要求信号を入力してください。

・CH0, CH1の同時取込例

・CH0, CH1, CH2, CH3の同時取込例

・関連ライブラリ

Lib_freeze
Lib_set_video_input_mode
Lib_get_video_input_mode
Lib_xfreeze
Lib_get_video_status
Lib_set_sim_channel
Lib_set_sim_gray_memory
Lib_set_sim_bin_memory
Lib_set_exposure

2.5　FVC02について

■サポートカメラ
2004.04.01現在のサポートカメラは以下のようになります。

カメラ	メーカー
XC-55	ソニー製
XC-HR300	ソニー製
XC-HR50	ソニー製
XC-HR70	ソニー製(※)
CS8530-01	東芝テリー製
CS8530D-01	東芝テリー製
CS3720	東芝テリー製
CS3730DC	東芝テリー製
CS8550D(CS8550Di)	東芝テリー製
CS8550i-01	東芝テリー製
TI-480A	日興電気通信製
TI-1200A	日興電気通信製(※)
EIA-170	EIA-170仕様カメラ

(※)FVL/DOS, 902ではサポートされていません。

■2枚搭載時のチャネル割付
FVC02が2枚搭載されている場合のチャネルは、以下の表のようになります。

搭載番号	チャネル番号
1枚目	CH0, 1
2枚目	CH2, 3

■同時取込
FVC02 は同時取込をサポートしますが、ビデオ入力モード、使用カメラにより以下の注意事項があります。

入力モード	注意事項
通常入力	同一ボード内の2CHのみ同時取込可能。
内部ランダムトリガ	任意の2CH～4CH同時取込可能。 ボード間は同期連動します。
外部ランダムトリガ	任意の2CH～4CH同時取込可能。 同じボード内の取込要求 TRGRQ0、TRGRQ1は同期連動します。 いずれか1つの信号を入力してください。 異なるボード間の取込要求は同期連動しません。 必ずそれぞれのボードの取込要求信号を入力してください。

カメラ	注意事項
XC-55	2CH～4CH同時取込可能。
XC-HR300	同時取込不可。
XC-HR50	2CH同時取込可能。
XC-HR70	同時取込不可。
CS8530-01	2CH～4CH同時取込可能。
CS8530D-01	2CH同時取込可能。
CS3720	同時取込不可。
CS3730DC	同時取込不可。
CS8550D(CS8550Di)	2CH同時取込可能。
CS8550i-01	2CH～4CH同時取込可能。
TI-480A	同時取込不可。
EIA-170	2CH同時取込可能。
TI-1200A	同時取込不可。

■フリーラン
FVL/DOS, 902では、XC-HR300, CS8530D-01, CS3720, TI-480A, XC-HR50, CS3730DC, CS8550D(CS8550Di)
では、フリーラン動作はできません。

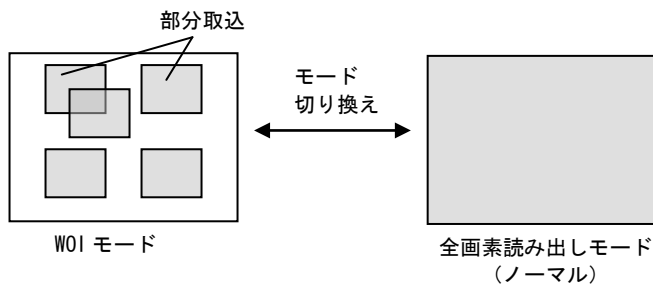
2.6　FVC04について

■サポートカメラ（FVL/LNXのみ対応）
2004.02.09現在のサポートカメラは以下のようになります。

カメラ	メーカー
CSB4000CL-10	東芝テリー製
CSB1100CL	東芝テリー製
KP-F100BCL	日立国際電気製
KP-F120CL	日立国際電気製
IK-SX1	東芝製

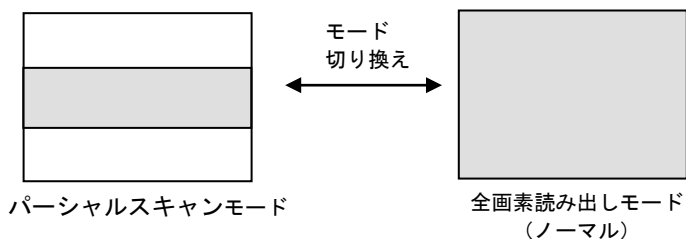
■カメラ固有の機能

1.WOI機能

東芝テリー製CSB1100CL及びCSB4000CLカメラが持つWOI(Window of Interest)機能を使用することにより任意の複数エリア(最大16)を設定し取り込むことが可能です。弊社では、以下のようにモードを切り換えることでこの機能をサポートしています。

2.マルチスロープ機能

東芝テリー製CSB1100CL 及びCSB4000CLカメラが持つマルチスロープ機能を使用して広ダイナミックレンジの画像を得ることができます。

3.パーシャルスキャン機能

日立国際電気製KP-F120CL、東芝製IK-SX1において以下のようにモードを切り換えることで、パーシャルスキャン機能をサポートします。

2.7 FVC05について

■サポートカメラ （FVL/DOS, 902, FVL/LNXのみ対応）
2008.02.01現在のサポートカメラは以下のようになります。

カメラ	メーカー	補足
XC-HR50/XC-HR57	ソニー製	60F HR50:1/3, HR57:1/2
XC-55/XC-56	ソニー製	30F （※1）
CS8550D/CS8550Di	東芝テリー製	60F 1/3
CS8560D/CS8570D	東芝テリー製	60F CS8560D:1/3 CS8570D:1/2（※2）
KP-F30	日立国際電気製	60F 1/3（※2）
KP-F33	日立国際電気製	30F 1/3（※2）
IK-52V/53V	東芝製	60F IK-52V:1/2 IK-53V:1/3（※2）
TI-480A	日興電気通信製	60F 1/3
EIA-170 （外部同期）	EIA-170仕様カメラ	30F
以下、FVL/LNXのみ対応		
XC-HR70	ソニー製	30F 1024x768, 992x768, 512x480
CS8560D1/CS8570D（パーシャル）	東芝テリー製	1/2, 1/4
CS8550i-01	東芝テリー製	30F 640x480, 512x480(※1)
CS8550i-51	東芝テリー製	30F 640x480, 512x480(※1)
CV-A1	JAI製	30F 1376x1040
CV-A2	JAI製	15F 1600x1200
STC-A33A	SENTECH製	60F 512x480
STC-A33A	SENTECH製	90F 512x480
STC-A33A（パーシャル）	SENTECH製	1/2, 1/4
STC-A83A	SENTECH製	30F 1024x768
STC-A83A（パーシャル）	SENTECH製	1/2, 1/4
STC-A152A	SENTECH製	15F 1344x1024
STC-A152A(パーシャル)	SENTECH製	1/2, 1/4
STC-A202A	SENTECH製	15F 1600x1200
STC-A202A(パーシャル)	SENTECH製	1/2, 1/4

(※1)既存のFVC05では12ピンケーブルのピン配置が異なるため、画像を入力することができません。
そのため、FVC05のピン配置を変換するブラケットが必要になります。
詳しくは、高速アナログ画像入力ボード FVC05取扱説明書を参照下さい。
(※2)FVL/LNXでは取込画像サイズが640x480になります。

■2枚搭載時のチャネル割付
FVC05が2枚搭載されている場合のチャネルは、以下の表のようになります。

搭載番号	チャネル番号
1枚目	CH0, 1
2枚目	CH2, 3

■同時取込

FVC05 は同時取込をサポートしますが、ビデオ入力モード、使用カメラにより以下の注意事項があります。

入力モード	注意事項
通常入力	同一ボード内の2CHのみ同時取込可能。
内部ランダムトリガ	任意の2CH～4CH同時取込可能。 ボード間は同期連動します。 （トリガ連動ケーブルの接続が必要）
外部ランダムトリガ	任意の2CH～4CH同時取込可能。 同じボード内の同時取込の場合、取込要求 TRGRQ0, TRGRQ1 はいずれか一つの信号を入力して下さい。 ボードを跨いだ同時取込の場合は必ずマスタ側のボード(CH0側のボード)に取込要求信号を入力して下さい。(トリガ連動ケーブルの接続が必要)

カメラ	注意事項	補足
XC-HR50/XC-HR57	2CH～4CH同時取込可能。	60F HR50:1/3, HR57:1/2
CS8550D/CS8550Di	2CH～4CH同時取込可能。	60F 1/3
CS8560D/CS8570D	2CH～4CH同時取込可能。	60F CS8560D:1/3 CS8570D:1/2
KP-F30	2CH～4CH同時取込可能。	60F 1/3
KP-F33	2CH～4CH同時取込可能。	30F 1/3
IK-52V/53V	2CH～4CH同時取込可能。	60F IK-52V:1/2 IK-53V:1/3
TI-480A	同時取込不可。	60F 1/3
EIA-170（外部同期）	2CH同時取込可能。	30F

■フリーラン

FVL/DOS, 902では、60F以上のカメラはフリーラン動作できません。

2.8 FV1100について

■サポートカメラ
2004.09.01 現在の対応状況は以下のようになります。

カメラ	メーカー	推 奨	備 考
XC-HR70	ソニー製	○	
XC-HR50	ソニー製	○	
CS8550Di	東芝テリー製	○	
IK-52(53)V	東芝製	○	
KP-F30	日立国際電気	○	
EIA170	EIA170仕様カメラ	○	
CV-A1	ジェイエイアイコーポレーション		※1
TI-480A	日興電気通信製		※2
TI-1250A	日興電気通信製		※2

※1 取込方法を切替えるためには専用のシリアル接続ケーブルが必要となります。
※2 同時取込はノーマル取込のみとなります。

2.9 FVC06について

■サポートカメラ
2008.09.01 現在の対応状況は以下のようになります。

カメラ	メーカー	画像サイズ	備 考
CSB1100CL	東芝テリー製	1216x1022	※1
CSB4000CL	東芝テリー製	1984x2016	※1
CS3970CL	東芝テリー製	1600x1236	
CV-M2CL	JAI製	1600x1200	
VCC-G223V31CL	CIS製	640x480	
KP-F200SCL-S1	日立国際電気製	1600x1236	
KP-F31SCL	日立国際電気製	640x480	
KP-F80SCL	日立国際電気製	1024x768	
KP-F200SCL	日立国際電気製	1600x1236	
KP-F230SCL	日立国際電気製	1600x1236	
KP-F500SCL	日立国際電気製	2432x2058	
TL-7400RCL	竹中システム機器製	7392画素	ラインセンサ
NSUF0514	NED製	512画素	ラインセンサ

※1 WOI 機能には対応していません。

■2 枚搭載時のチャネル割付
FVC06 が 2 枚搭載されている場合のチャネルは、以下の表のようになります。

搭載番号	チャネル番号
1枚目	CH0, 1
2枚目	CH2, 3

2.10 FV-GP440について

■サポートカメラ(FVL/LNXのみ対応)
2009.05.01現在のサポートカメラは以下のようになります。

カメラ	メーカー	補　足
XC-HR50/XC-HR57	ソニー製	60F HR50:1/3, HR57:1/2
XC-HR70	ソニー製	30F 1024x768, 992x768, 512x480
XC-HR90	ソニー製	30F/60F 1280x960
XC-55/XC-56	ソニー製	30F
CS8550D/CS8550Di	東芝テリー製	60F 1/3
CS8550i-01	東芝テリー製	30F 640x480, 512x480
CS8550i-51	東芝テリー製	30F 640x480, 512x480
CS8560D/CS8570D	東芝テリー製	60F CS8560D:1/3 CS8570D:1/2
CS8560D1/CS8570D	東芝テリー製	1/2, 1/4（パーシャル）
KP-F30	日立国際電気製	60F 1/3
KP-F33	日立国際電気製	30F 1/3
IK-52V/53V	東芝製	60F IK-52V:1/2 IK-53V:1/3
TI-480A	日興電気通信製	60F 1/3
EIA-170（外部同期）	EIA-170仕様カメラ	30F
CV-A1	JAI製	30F 1376x1040
CV-A2	JAI製	15F 1600x1200
STC-A33A	SENTECH製	60F 512x480
STC-A33A	SENTECH製	90F 512x480
STC-A33A（パーシャル）	SENTECH製	1/2, 1/4
STC-A83A	SENTECH製	30F 1024x768
STC-A83A（パーシャル）	SENTECH製	1/2, 1/4
STC-A152A	SENTECH製	15F 1344x1024
STC-A152A(パーシャル)	SENTECH製	1/2, 1/4
STC-A202A	SENTECH製	15F 1600x1200
STC-A202A(パーシャル)	SENTECH製	1/2, 1/4

■2枚搭載時のチャネル割付
FV-GP440 が2枚搭載されている場合のチャネルは、以下の表のようになります。

搭載番号	チャネル番号
1枚目	CH0, 1, 2, 3
2枚目	CH4, 5, 6, 7

■同時取込
FV-GP440 は同時取込をサポートしますが、ビデオ入力モードにより以下の注意事項があります。

入力モード	注意事項
通常入力	同一ボード内の 4CH のみ同時取込可能。
内部ランダムトリガ	任意の 2CH～8CH 同時取込可能。 ボード間は同期連動します。 （トリガ連動ケーブルの接続が必要）
外部ランダムトリガ	任意の 2CH～8CH 同時取込可能。 同じボード内の同時取込の場合、取込要求 TRGRQ0, TRGRQ1, TRGRQ2, TRGRQ3 はいずれか一つの信号を入力して下さい。 ボードを跨いだ同時取込の場合は必ずマスタ側のボード(CH0 側のボード)に取込要求信号を入力して下さい。 （トリガ連動ケーブルの接続が必要）

2.11　FHC3312について

■サポートカメラ(FVL/LNXのみ対応)
2009.08.31現在のサポートカメラは以下のようになります。

カメラ	メーカー	補　足
BM-500CL	JAI製	15fps 2432x2058 500万画素
VCC-G60FV11CL	CIS製	16fps 2432x2058 500万画素
CSCU30BC18	東芝テリー製	30fps 1600x1236 200万画素
CSC12M25BMP19	東芝テリー製	25fps 4096x3072 1258万画素
KP-F31SCL	日立国際電気製	120fps 640x480 33万画素
KP-F230SCL	日立国際電気製	30fps 1600x1236 200万画素
NUSF0514	NED製	ラインセンサ 512画素
NSUF1014	NED製	ラインセンサ 1024画素
SUFi74	NED製	ラインセンサ 7400画素

■2枚搭載時のチャネル割付
FHC3312 が2枚搭載されている場合のチャネルは、以下の表のようになります。

搭載番号	チャネル番号
1枚目	CH0, 1
2枚目	CH2, 3

■同時取込
FHC3312 は同時取込をサポートしますが、ボードを跨ぐ同時取込はサポートされていません。

2.12　FHC3313について

■サポートカメラ(FVL/LNXのみ対応)
2009.08.31現在のサポートカメラは以下のようになります。

カメラ	メーカー	補　足
PIRANHA3	DALSA製	ラインセンサ 12288画素
CSC12M25BMP19	東芝テリー製	25fps 4096x3072 1258万画素

FHC3313の搭載可能枚数は1枚までです。

Lib_freeze

機　能

ビデオ フリーズ

形　式

```
#include "f_video.h"
int Lib_freeze( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

最新の入力ビデオをカレントメモリ上に凍結します。
フリーラン中に起動した場合は、モニタ表示は静止画状態となります。
フリーズ中に起動した場合は、モニタ表示は変わりません。
画像メモリとモニタ表示を一致させたい場合は、フレームバッファへの画像転送指示をしてください。

① ***flag*** はフレームバッファへの画像転送フラグです。

値	定　数	意　味
1	TRANSMIT	フレームバッファへ画像を転送します。
0	NOT_TRANSMIT	フレームバッファへ画像を転送しません。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

①ビデオチャネル0(オフセット=0, 2値化レベル=128)から2値画像の入力をします。

```
#include "f_video.h"

#define  SET_CHANNEL   0
#define  SET_BINARY    128
#define  SET_OFFSET    0

void set_freeze()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE );

    Lib_freeze( TRANSMIT );
}
```

②ビデオチャネル0(オフセット=0, 2値化レベル=128)から2値画像を入力し、前処理後、画像を表示します。

```
#include "f_video.h"

#define  SET_CHANNEL  0
#define  SET_BINARY   128
#define  SET_OFFSET   0

void set_freeze()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE );

    Lib_freeze( NOT_TRANSMIT );

    /* 前処理 */

    Lib_video_transmit( BIN_PLANE );
}
```

留意事項

共通

○ビデオ入力用現在チャネル(システムパラメータ)
システムパラメータ「ビデオ入力用現在チャネル」により、ビデオデータを入力するチャネルが指定されます。Lib_channel, Lib_video_channel, Lib_stage_input, Lib_set_stage_input により、指定できます。

○ビデオ入力オフセット値(システムパラメータ)
システムパラメータ「ビデオ入力オフセット値」にて入力ビデオのオフセット方式が指定されます。Lib_channel, Lib_video_offset により、指定できます。

○ビデオ入力制御(システムパラメータ)
システムパラメータ「ビデオ入力制御」にて入力ビデオの映像が指定されます。
Lib_input_video_control により、入力映像を指定できます。

○メモリ
☆濃淡画像メモリ8ビット全体に取込まれます。
☆2値画像1ビット、1画素にビットマッピングされ取り込まれます。

○画像の転送
画像の転送を行わない場合、表示画面(フレームバッファ)とメモリ内容は違います。
すなわち画像変形処理を行った場合、その画像は改めて転送しないと表示されません。

○画像転送フラグ、入力ビデオ制御(Lib_input_video_control)により所要時間が変わります。
実行所要時間代表値については、付録を参照してください。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

901

〇画像転送フラグを NOT_TRANSMIT にした場合のモニタTV表示内容

☆フリーラン中の起動

モニタTV表示画面は画像メモリ内容の1フレーム前のものとなります。

☆フリーズ中の起動

モニタTV表示画面はかわりません。

〇割り込み

画像入力中はブレーク割り込み、V割り込み以外は禁止されています。

902

〇画像転送フラグを NOT_TRANSMIT にした場合のモニタTV表示内容

☆フリーラン中の起動

VCAP, CATS　　　　：最大で1フレーム前後します。

RICE, FVC01, FVC02, FVC05：画像メモリ内容の1フレーム前のものとなります。

☆フリーズ中の起動

モニタTV表示画面はかわりません。

903, 904

〇画像転送フラグを NOT_TRANSMIT にした場合のモニタTV表示内容

☆フリーラン中の起動

モニタTV表示画面は画像メモリ内容の1フレーム前のものとなります。

☆フリーズ中の起動

モニタTV表示画面はかわりません。

FVL/LNX

〇画像転送フラグを NOT_TRANSMIT にした場合のモニタTV 表示内容

☆フリーラン中の起動

モニタTV表示画面は画像メモリ内容の1フレーム前のものとなります。

☆フリーズ中の起動

モニタTV表示画面は変わりません。

〇フリーラン中に起動した場合、本関数の所要時間にフリーランを停止する所要時間が加算されます。

Lib_freerun

機　能

ビデオ フリーラン

形　式

```
#include "f_video.h"
int Lib_freerun(void);
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

フレームバッファ内容を新規入力ビデオにより逐次更新します。
以降、新しい入力ビデオによりリフレッシュされます。画像メモリには取り込まれません。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル0(オフセット=0, 2値化レベル=128)から2値画像の入力をします。

```
#include "f_video.h"

#define  SET_CHANNEL  0
#define  SET_BINARY   128
#define  SET_OFFSET   0

void set_freerun()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE );

    Lib_freerun( );
}
```

留意事項

共通

○ビデオ入力用現在チャネル(システムパラメータ)
システムパラメータ「ビデオ入力用現在チャネル」により、ビデオデータを入力するチャネルが指定されます。
Lib_channel, Lib_video_channel, Lib_stage_input, Lib_set_stage_input により、指定できます。

○ビデオ入力オフセット値(システムパラメータ)
システムパラメータ「ビデオ入力オフセット値」にて入力ビデオのオフセット方式が指定されます。
Lib_channel, Lib_video_offset により、指定できます。

○ビデオ入力制御(システムパラメータ)
システムパラメータ「ビデオ入力制御」にて入力ビデオの映像が指定されます。
Lib_input_video_control により、入力映像を指定できます。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

902

○FVC01、FVC02、FVC05で2倍速以上のカメラ使用時は、フリーラン動作出来ません。

○2倍速のカメラ
・XC-7500　　　(ソニー製)
を使用してのフリーランでは、フリーラン中にモニタTVに描画を行うと、表示画像が乱れる場合があります。特に連続して描画を行う場合にこの現象が顕著に表れますが、これらは異常ではありません。
この現象が問題となる場合は、描画をフリーズ状態で行ってください。
この現象は、画像処理で使用するためにメモリに取込まれた画像には影響しません。

FVL/LNX

○90X ではフリーラン時に CPU をほぼ開放しておりましたが、FVL/LNX ではフリーラン中にもある程度 CPU の計算能力を消費します。
よってフリーラン中に何らかの動作を行うと、CPU を 100%占有することは出来ません。
CPU を 100%占有して計算などを行いたい場合は Lib_stop_freerun( )関数にてフリーランを停止してから計算を実行するようにして下さい。
また、フリーラン停止は 100～200(msec)程度の処理時間がかかります。連続して処理を行う場合などは、極力フリーラン状態に移行しない方が良いと思われます。

○フリーラン中にRS232Cにて高速通信(115200bps等)を行うと通信に失敗する事があります。その場合は、

・フリーランを停止してから通信を行う
・通信速度を下げて通信を行う

などの対処を行ってください。

Lib_stop_freerun

機　能

ビデオフリーラン停止

形　式

```
#include "f_video.h"
int Lib_stop_freerun( void );
```

901	902	903	904	FVL/LNX	高分解能
				○	△

解　説

フリーランを停止します。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

フリーランを停止後、連続して処理を行います。

```
void func( void )
{
    int    x, y;

    Lib_stop_freerun();
    for( ; ; )
    {
            Lib_freeze();

            /* 何らかの処理 */

            if( 0 != Lib_see_current_position_real(&x,&y) )
                    break;
    }
}
```

留意事項

○FVL/LNXではフリーラン中であってもCPUによる処理を行っています。
　タクトを重視する動作を行う場合は、極力フリーランを停止するようにして下さい。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_get_freerun_status

機　能

フリーラン状態取得

形　式

```
#include "f_video.h"
int Lib_get_freerun_status( void );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

フリーラン状態を取得します。

戻り値

処理結果

値	定　数	意　味
0		フリーランは停止しています。
0以外		フリーランは動作中(準備中)です。

例

ありません。

留意事項

ありません。

Lib_clear_freerun_buffer

機　能　フリーランバッファクリア

形　式

```
#include "f_video.h"
void Lib_clear_freerun_buffer( void );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　フリーランバッファをクリアします。

戻り値　ありません。

例　ありません。

留意事項　ありません。

Lib_set_freerun_scale

機　能

フリーランの表示倍率設定

形　式

```
#include "f_video.h"
int Lib_set_freerun_scale( int scale );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

フリーラン表示倍率を設定します。

① ***scale*** は表示倍率です。

値　　定　　数	意　　味
0 ＜ scale	フリーランの表示をscale倍に設定します。 （拡大表示）
0 ＞ scale	フリーランの表示をscale分の1に設定します。 （縮小表示）

戻り値

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-10002	ERR_ILLEGAL_PARAM	scaleに0が設定されました。

例

SDK付属のサンプルソフトのfreerun_scaleを参照ください。

```
// freerunの表示倍率を2倍にします
int sample1( void )
{
        Lib_set_freerun_scale( 2 );
}


// freerunの表示倍率を1/2倍にします
int sample1( void )
{
        Lib_set_freerun_scale( -2 );
}
```

留意事項

○windowのサイズは変更されませんので、windowをカメラサイズ分に設定している場合には、フリーラン画像の全てがwindowに表示されるわけではありません。

Lib_get_freerun_scale

機　能

フリーランの表示倍率取得

形　式

```
#include "f_video.h"
int Lib_get_freerun_scale( void );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

フリーラン表示の縮小率を取得します。

戻り値

フリーランの表示倍率

値　定　数	意　味
0 < 戻り値	フリーランの表示はscale倍に設定されています。　（拡大表示）
0 > 戻り値	フリーランの表示はscale分の1に設定されています。（縮小表示）

例

留意事項

ありません。

Lib_freeze_tm9700

機　能

フルフレーム・シャッターカメラ映像入力

形　式

```
#include "f_video.h"
int Lib_freeze_tm9700( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○			○		

解　説

フルフレーム・シャッターカメラTM-9700/AD-10(竹中システム機器製)を使用して、ランダムトリガでのフルフレーム画像を、カレントメモリ上に取込ます。

① ***flag*** はフレームバッファへの画像転送フラグです。

値	定　　数	意　　味
0	NOT_TRANSMIT	フレームバッファへ、取り込んだ画像を転送しません。
1	TRANSMIT	フレームバッファへ、取り込んだ画像を転送します。

戻り値

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例　トラックボール左ボタン押下で画像を入力します。

```
#include "f_gui.h"              /* GUI        */
#include "f_video.h"            /* ビデオ入力制御     */
#include "f_time.h"             /* 時刻サービス*/

void main( void )
{

        int cpx,cpy;
        int c_status;

        Lib_init_cursor();

        Lib_input_video_control( GRAY_PLANE );
        Lib_display_control( GRAY_PLANE | LINE_PLANE | CHAR_PLANE );
        Lib_set_field_squence( SET_ANY_FIELD );
        Lib_set_strobo( TTL_TYPE, SET_ANY_FIELD );
        Lib_ctrl_strobo( START_STROBO );

        for ( ; ; )
        {

                if ( CURSOR_EXECUTE == ( c_status = Lib_see_current_position
                                                  ( &cpx, &cpy ) ) )
                {

                        Lib_ctrl_strobo( STOP_STROBO );
                        Lib_time_delay( 33 );
                        Lib_strtclk_count();
                        Lib_freeze_tm9700( TRANSMIT );

                        Lib_display_keyinput( 0, 0, "確認" );
                }
                else if ( CURSOR_CANCEL == c_status )
                        break;

                Lib_freerun();
                Lib_ctrl_strobo( START_STROBO );

        }

        Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );

}
```

留意事項

○映像入力のフィールド・シーケンスは、Lib_set_field_sequenceライブラリを使用して、
「任意」または「第2フィールドから」に設定しておいてください。

○必ず実時間クロック(微細クロックモード)をLib_strtclk_countライブラリを使用して、
スタートさせておいてください。
(Lib_strtclk_count に関しては、「90X基本ライブラリ説明書 Vol.2」をご参照下さい。)

○STROBO信号コネクタ(装置背面パネル)と、AD-10の
C.VIDEO(IN)コネクタを、BNCケーブルで接続してください。
(カメラへのトリガ信号として使用)
ストロボ出力の設定は、Lib_set_stroboライブラリを使用してプルアップ出力に設定してください。

○AD-10のSHUTTERスイッチは、OFFに設定してください。

○シャッタ速度は、TM-9700背面のダイアルスイッチで設定してください。
(FAST_RESET/SLOW_RESET モードが使用できます)

○ビデオ入力モードは、「通常入力モード」に設定しておいてください。
(日興電気通信製TI-324Aのランダムトリガモードとは異なりますので御注意下さい)

○フレームバッファの動画表示には、通常のLib_freerunライブラリを使用してください。

○その他留意事項は、Lib_freezeライブラリを参照してください。

○本ライブラリはFV904p, FV904pcではサポートしておりません。

Lib_strobo_freeze

機　能

ストロボ発光映像入力

形　式

```
#include "f_video.h"
int  Lib_strobo_freeze( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	△		△		

解　説

最新の入力ビデオをストロボ出力後、カレントメモリ上に凍結します。
すなわち、現在フリーラン中の場合、ストロボ端子よりトリガ出力後、画像メモリに入力ビデオが取り込まれ、フレームバッファへのビデオ流入がストップします。
つまり表示は動画状態から取り込まれ画像の静止状態となります。
現在フリーズ中の場合、ストロボ端子よりトリガ出力後、画像メモリに入力ビデオが取り込まれ、フレームバッファに取り込まれた静止画が表示されます。

① ***flag*** はフレームバッファへの画像転送フラグです。

値	定　　数	意　　味
1	TRANSMIT	フレームバッファへ画像を転送します。
0	NOT_TRANSMIT	フレームバッファへ画像を転送しません。

戻り値

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ストロボ入力のサンプルプログラムを参照してください。

留意事項

共通

○映像の入力は常に任意のフィールドから2フィールド入力します。

○CCDカメラは、フレーム蓄積モードに設定してください。
　設定方法は、各カメラの操作説明書を参照してください。

○初期設定値は出力信号がオープンコレクタ出力で入力タイミングは自由タイミングとなっていますのでそれ以外の設定は後頁の Lib_set_strobo で設定してください。

○Lib_freezeの留意事項も参照してください。

901, 904

○Lib_set_stroboを実行後は、1フレーム期間(33ms)以上たってから本関数を実行してください。

○本ライブラリはFV904p、FV904pcではサポートしておりません。

902

○本ライブラリはVCAP, FVC01, FVC02, FVC05ではサポートしておりません。

Lib_set_strobo

機　能

ストロボの設定

形　式

```
#include "f_video.h"
void Lib_set_strobo( int ctrl, int field );
```

901	902	903	904	FVL/LNX	高分解能
○	△		△		

解　説

ストロボの出力の設定を行います。

① ***ctrl*** はストロボ出力方法です。

値	定　　数	意　　味
0	O_C_TYPE	オープンコレクタ出力を指定します。
1	TTL_TYPE	プルアップ出力を指定します。

② ***field*** は入力モードです。

値	定　　数	意　　味
0	SET_ANY_FIELD	Lib_strobo_freeze をコールするとノーウエイトでパルス出力をします。
1	SET_1ST_FIELD	第1フィールド先頭に同期パルス出力を指定します。
2	SET_2ND_FIELD	第2フィールド先頭に同期パルス出力を指定します。
3	SET_BOTH_FIELD	両フィールドの先頭に同期パルス出力を指定します。

戻り値

ありません。

例

①カメラの絞りを合わせた後、ストロボ発光映像入力を実行します。(901用)

```
#include "f_gui.h"
#include "f_video.h"

void  strobo()
{
    Lib_init_cursor();
    Lib_set_strobo( O_C_TYPE, SET_1ST_FIELD );
    Lib_ctrl_strobo( START_STROBO );
    Lib_freerun();
    Lib_display_keyinput( 10, 450, "入力待ち");
    Lib_ctrl_strobo( STOP_STROBO );
    Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );
    Lib_time_delay(33);
    Lib_strobo_freeze( TRANSMIT );
}
```

②ストロボ発光映像入力を実行します。(902用)

```
#include "f_gui.h"
#include "f_video.h"


void main( void )
{
    Lib_init_cursor();      /* カーソル操作のための初期化 */

    Lib_set_strobo( O_C_TYPE, SET_ANY_FIELD );    /* 設定 */
    Lib_set_video_input_mode( STROBO_VIDEO_MODE, 0 );

    Lib_freeze( TRANSMIT );
}
```

留意事項

共通

〇Lib_strobo_freezeを実行する前に必ず入力モードは ”SET_ANY_FIELD” に戻してください。

901

〇本関数で設定後、1フレーム期間(33ms)以上たってからLib_strobo_freezeを実行してください。

902

〇CATS, RICEは、入力モードの1～3についてはサポートしておりません。

〇本ライブラリは VCAP, FVC01, FVC02, FVC05 ではサポートしておりません。

〇CATS, RICE は、チャネルごとにストロボの設定が可能です。本ライブラリでのストロボ設定は、カレントビデオ入力チャネルに対し設定されます。

904

〇本関数で設定後、1フレーム期間(33ms)以上たってからLib_strobo_freeze を実行してください。

〇入力モードの1～3についてはサポートしておりません。

〇本ライブラリはFV904p、FV904pcではサポートしておりません。

Lib_ctrl_strobo

機　能

ストロボの制御

形　式

```
#include "f_video.h"
void Lib_ctrl_strobo( int ctrl );
```

901	902	903	904	FVL/LNX	高分解能
○			○		

解　説

フリーラン中のストロボ発光の制御を行います。

① ***ctrl*** はストロボ発光のモードです。

値	定　数	意　味
1	START_STROBO	ストロボ動作開始を指定します。
0	STOP_STROBO	ストロボ動作停止を指定します。

戻り値

ありません。

例

カメラの絞りを合わせた後、ストロボ発光映像入力を実行します。

```
#include "f_gui.h"
#include "f_video.h"

void  strobo()
{
    Lib_init_cursor();
    Lib_set_strobo( O_C_TYPE, SET_1ST_FIELD );
    Lib_ctrl_strobo( START_STROBO );
    Lib_freerun();
    Lib_display_keyinput( 10, 450, "入力待ち");
    Lib_ctrl_strobo( STOP_STROBO );
    Lib_set_strobo( O_C_TYPE, SET_ANY_FIELDF );
    Lib_time_delay(33);
    Lib_strobo_freeze( TRANSMIT );
}
```

留意事項

○本ライブラリはFV904p、FV904pcではサポートしておりません。

Lib_set_field_squence

機　能

フィールド・シーケンスの設定

形　式

```
#include "f_video.h"
int Lib_set_field_squence( int mode );
```

901	902	903	904	FVL/LNX	高分解能
△	△		△		

解　説

映像入力のフィールド・シーケンスを設定します。
初期設定状態は任意のフィールドから2フィールド入力します。

① ***mode*** は入力モードです。

値	定　　数	意　　味
0	SET_ANY_FIELD	任意のフィールドから2フィールド入力します。
1	SET_1ST_FIELD	第1フィールドから2フィールド入力します。
2	SET_2ND_FIELD	第2フィールドから2フィールド入力します。
3	SET_ANY_FIELD_CONTINUOUS	任意のフィールドから連続フレーム入力します。
4	SET_ANY_SINGLE_FIELD	任意のフィールドから1フィールド入力します。
5	SET_1ST_SINGLE_FIELD	第1フィールドから1フィールド入力します。
6	SET_2ND_SINGLE_FIELD	第2フィールドから1フィールド入力します。

戻り値

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_video.h"

void input_video()
{
    if ( NORMAL_RETURN == Lib_set_field_squence( SET_1ST_FIELD ) )
    {
        Lib_freeze( TRANSMIT );      /* ビデオ フリーズ */
    }
}
```

留意事項

共通

〇初期設定状態の任意のフィールドから2フィールド入力した場合の処理時間は
33～49msですが、mode = 1 or 2 の場合は 33～66msかかります。

〇1フィールド入力(mode = 4 or 5 or 6)する場合、ビデオフリーズは
Lib_freeze(NOT_TRANSMIT)にて行い、フレームバッファへの画像転送はしないでください。
また映像は画像メモリの上半分に取込まれます。
またカレントメモリを切替える場合は、本ライブラリより前に実行してください。

901, 904

〇任意のフィールドから連続フレーム入力（mode = 3）する場合、ビデオフリーズは
Lib_frame_chain_freeze, Lib_camera_chain_freeze もしくは、
Lib_xframe_chain_freeze, Lib_xcamera_chain_freeze を使用してください。

〇入力モードの3～6についてはCSC901, CSC901a, CSC901SBではサポートしておりません。

〇本ライブラリはFV904p、FV904pcではサポートしておりません。

902

〇入力モードの3についてはCATS、RICEではサポートしておりません。
RICEは、任意のフィールドから連続フレーム入力をサポートしていますが、入力モードは、
0を指定してください。

〇入力モードの1～6についてはVCAPではサポートしておりません。

〇本ライブラリは、FVC01, FVC02, FVC05ではサポートしておりません。

Lib_set_video_input_mode

機　能

ビデオ入力モード設定

形　式

```
#include "f_video.h"
int Lib_set_video_input_mode ( int mode, int channel );
```

901	902	903	904	FVL/LNX	高分解能
△	△	△	△	○	

解　説

指定したチャネル番号のビデオ入力モードを設定します。

①***mode*** は入力モードです。機種ごとに設定可能な値が異なります。

901

値	定　数	意　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x00000001	STROBO_VIDEO_MODE	ストロボ入力モードに設定します。
0x000000ff	RANDOM_VIDEO_MODE	ランダムトリガモードに設定します。
0x000fffff	RANDOM_FULLFRAME_VIDEO_MODE2	ランダムトリガモード（フルフレーム）に設定します。

904a，904ac

値	定　数	意　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x00000001	STROBO_VIDEO_MODE	ストロボ入力モードに設定します。
0x000fffff	RANDOM_FULLFRAME_VIDEO_MODE2	ランダムトリガモード（フルフレーム）に設定します。

904p，904pc

値	定　数	意　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x01000000	RANDOM_XC_MODE	ランダムトリガモードに設定します。

903

値	定　数	意　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x01000000	RANDOM_XC_MODE	ランダムトリガモードに設定します。

902 - RICE

値	定　　　数	意　　　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x00000001	STROBO_VIDEO_MODE	ストロボ入力モードに設定します。
0x000fffff	RANDOM_FULLFRAME_VIDEO_MODE2	ランダムトリガモード（フルフレーム）に設定します。
0x00001000	SIM_VIDEO_MODE	同時取込モードに設定します。
0x00001001	NORMAL_2N_MODE	2ライン取込ノンインターレースモードに設定します。
0x00001003	RANDOM_XC7500_2I_RESET_MODE	XC-7500 2ライン出力インターレース RESET モードに設定します。

902 - FVC01

値	定　　　数	意　　　味
0x00000000	NORMAL_VIDEO_MODE	通常入力モードに設定します。
0x01000000	RANDOM_XC_MODE	ランダムトリガモードに設定します。
0x01000002	RANDOM_XC_EX_MODE	外部ランダムトリガモードに設定します。
0x01000003	RANDOM_XC_SIM_MODE	同時ランダムトリガモードに設定します。
0x01000004	RANDOM_XC_EX_SIM_MODE	外部・同時ランダムトリガモードに設定します。

902 - FVC02

未サポートです。Lib_set_xvideo_input_modeをご使用下さい。

902 - FVC05

未サポートです。Lib_set_xvideo_input_modeをご使用下さい。

② ***channel*** はチャネル番号です。（0～3。ただし、901AT、903では0～1）

戻り値

処理結果

値	定　　　数	意　　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオ入力モードを参照し、ランダムトリガモードでない場合は変更します。

```
#include "f_video.h"

void random_channel( int channel )
{
    if( NORMAL_VIDEO_MODE == Lib_get_video_input_mode ( channel ))
    {
        Lib_set_video_input_mode( RANDOM_VIDEO_MODE, channel );
    }
    Lib_video_channel( channel );
}
```

留意事項

共通

○初期状態は通常入力モードです。

○この関数では各チャネルに対してのビデオ入力モードを設定するのみです。
処理を実行する場合は Lib_channel, または Lib_video_channelをコールする必要があります。
(903を除く)

○この関数で設定したモードは再設定するまで有効です。

○通常ビデオ入力モード設定チャネルとランダムトリガ・入力モード設定チャネルの変更は、
Lib_channel, またはLib_video_channelをコールするのみで行えます。
(903を除く)

○濃淡、2値メモリ共にメモリクリアを行っていません。ご留意下さい。

○Lib_random_video_channelは、互換性維持のために変更していません。

○Lib_channel, Lib_video_channel の処理時間に変更があります。
設定前チャネルと設定後チャネルのビデオ入力モードが違う場合最大33ms遅くなりました。
（ただし、ノーウエイトモード）

901, 904, 904a, 904ac

○使用可能なビデオ入力モードは以下のようになります。

ビデオ入力モード	901	904(a/ac)	備　　考
NORMAL_VIDEO_MODE	○	○	
STROBO_VIDEO_MODE	○	○	
RANDOM_VIDEO_MODE	○		1フィールド分の取込(TI-324A)
RANDOM_FULLFRAME_VIDEO_MODE2	○	○	CSC901NT, 904(a/ac) + IK-542のみ
SIM_VIDEO_MODE			
NORMAL_2N_MODE			
RANDOM_XC7500_2I_RESET_MODE			
RANDOM_XC_MODE			
RANDOM_XC_MODE_BOTH_CH			
RANDOM_XC_EX_MODE			
RANDOM_XC_SIM_MODE			
RANDOM_XC_EX_SIM_MODE			

○入力モードの RANDOM_FULLFRAME_VIDEO_MODE2 は、CSC901NT、FV904(a/ac)と
CCDカメラIK-542(東芝製)の1パルストリガ
SYNCノンリセットモード(1/30Iモード)との組み合わせのみ使用可能です。

904p, 904pc

ビデオ入力モード	904p/pc	備　　考
NORMAL_VIDEO_MODE	○	
STROBO_VIDEO_MODE		
RANDOM_VIDEO_MODE		1フィールド分の取込(TI-324A)
RANDOM_FULLFRAME_VIDEO_MODE2		CSC901NT, 904(a/ac) + IK-542のみ
SIM_VIDEO_MODE		
NORMAL_2N_MODE		
RANDOM_XC7500_2I_RESET_MODE		
RANDOM_XC_MODE	○	
RANDOM_XC_MODE_BOTH_CH		
RANDOM_XC_EX_MODE		
RANDOM_XC_SIM_MODE		
RANDOM_XC_EX_SIM_MODE		

○FV904p、FV904pcでは、丸型12pinコネクタのカメラに対応しています。サポートカメラにつきましては、前述の「2.3　904について」をご参照下さい。

○FV904p、FV904pcではXC-55/56で、ほぼカメラの中心部分が取り込まれる様にタイミングを設定しています。よってXC-55/56とタイミングの異なるカメラは、画像垂直取込ライン設定関数で調整が必要です。

XC-55/56とタイミングの異なるカメラ
CS8550i-01, CS8550D/CS8550Di　　・・・・・ +12
KP-F30, KP-F33　　・・・・・ -6

○XC-56互換カメラを使用する場合とそれ以外のカメラを使用する場合で904システム上で「トリガ出力ピン」の設定を変更する必要があります。初期値は「Disable」(無接続)となっていますので使用するカメラに合わせて最初に必ず1度は設定して下さい。設定後システムパラメータの保存(システムのメイン画面より○システム＞○セーブ)を実行すれば次回起動時より設定が有効となります。

○本関数は FVC02, FVC05ではサポートしていません。FVC02, FVC05の場合は、
Lib_set_xvideo_input_modeをお使い下さい。
○使用可能なビデオ入力モードは以下のようになります。

ビデオ入力モード	VCAP	CATS	RICE	FVC01	備　　考
NORMAL_VIDEO_MODE	○	○	○	○	
STROBO_VIDEO_MODE		○	○		
RANDOM_VIDEO_MODE		○			1フィールド分の取込(TI-324A)
RANDOM_FULLFRAME_VIDEO_MODE2		○	○		IK-542のみ
SIM_VIDEO_MODE	○		○		
NORMAL_2N_MODE			○		2ライン・ノンインターレースのみ
RANDOM_XC7500_2I_RESET_MODE			○		XC-7500のみ
RANDOM_XC_MODE				○	
RANDOM_XC_MODE_BOTH_CH					
RANDOM_XC_EX_MODE				○	
RANDOM_XC_SIM_MODE				○	
RANDOM_XC_EX_SIM_MODE				○	

○入力モードのRANDOM_FULLFRAME_VIDEO_MODE2は、CCDカメラIK-542(東芝製)
の1パルストリガSYNCノンリセットモード(1/30Iモード)との組み合わせのみ使用可能です。

○入力モードのSIM_VIDEO_MODEは、VCAP, RICEで同時取込を行う場合使用します。
VCAPでは、引数のチャネル番号は意味を持ちません。
RICEでは、チャネル番号に0または1を指定した場合は0と1チャネルの、2または3を指定した場合は2と3チャネルの2チャネル同時取込を行います。

○入力モードのNORMAL_2N_MODEは、2ライン出力、ノンインターレースCCDカメラを使用する場合に指定します。代表的なカメラとしては、XC-7500(ソニー製)があります。

○入力モードの RANDOM_XC7500_2I_RESET_MODE は、CCDカメラXC-7500(ソニー製)の2ライン出力インターレースRESETモードとの組み合わせのみ使用可能です。

○入力モードの RANDOM_XC_MODE は、CCDカメラXC-55(ソニー製)または、
CS8530-01(東芝テリー製)との組み合わせのみ使用可能です。

○入力モードの RANDOM_XC_EX_MODE は、CCDカメラXC-55(ソニー製)または、
CS8530-01(東芝テリー製)との組み合わせで、外部トリガ信号によるランダムトリガ取りこみを行う場合使用します。

○入力モードの RANDOM_XC_SIM_MODE は、CCDカメラXC-55(ソニー製)または、
CS8530-01(東芝テリー製)との組み合わせで、同時ランダムトリガ取りこみを行う場合使用します。このモードは全チャネルに設定されます。
また、2値のみの同時取込みはサポートしていません。

○入力モードの RANDOM_XC_EX_SIM_MODE は、CCDカメラXC-55(ソニー製)または、
CS8530-01(東芝テリー製)との組み合わせで、外部トリガ信号による同時ランダムトリガ取りこみを行う場合使用します。このモードは全チャネルに設定されます。
また、2値のみの同時取込みはサポートしていません。

903

○使用可能なビデオ入力モードは以下のようになります。

ビデオ入力モード	903	備　　考
NORMAL_VIDEO_MODE	○	
STROBO_VIDEO_MODE		
RANDOM_VIDEO_MODE		1フィールド分の取込(TI-324A)
RANDOM_FULLFRAME_VIDEO_MODE2		CSC901NT + IK-542のみ
SIM_VIDEO_MODE		
NORMAL_2N_MODE		
RANDOM_XC7500_2I_RESET_MODE		
RANDOM_XC_MODE	○	
RANDOM_XC_MODE_BOTH_CH		Lib_set_both_modeにて設定可
RANDOM_XC_EX_MODE		
RANDOM_XC_SIM_MODE		
RANDOM_XC_EX_SIM_MODE		

○RANDOM_XC_MODE_BOTH_CH モードは Lib_set_video_input_mode で設定することはできません。
Lib_set_both_mode にて設定します。

○切り替え時間は以下の通りです。

NORMAL_VIDEO_MODE　→　RANDOM_XC_MODE　約 220ms
RANDOM_XC_MODE　→　NORMAL_VIDEO_MODE　約 380ms
RANDOM_XC_MODE_BOTH_CH　→　NORMAL_VIDEO_MODE　約 380ms

HipsVer1.3 以降の場合、ライブラリ内で切り替え時間を待ちますが、HipsVer1.3 以前の 903 に関しましては切り替え後、画像取込(Lib_freeze 等)実行までには上記の時間を置く必要があります。

○入力モードが全チャネルに対して設定されます。(従って、チャネル番号パラメータは無視されます。)

FVL/LNX

○本来であればFVL/LNXで有効なビデオモードは、以下に示された3種類のみとなります。

NORMAL_XVIDEO_MODE　:　通常入力モード
RANDOM_XVIDEO_MODE　:　ランダムトリガモード
RANDOM_EX_XVIDEO_MODE　:　外部ランダムトリガモード

ただし、90Xで本関数を使用したアプリケーションが非常に多いことを考慮して、本関数で使用するビデオモードを内部で上記3種類のビデオモードに変換する形で、互換性を維持いたしました。

ビデオ入力モード	変換後のビデオ入力モード
NORMAL_VIDEO_MODE	NORMAL_XVIDEO_MODE
STROBO_VIDEO_MODE	NORMAL_XVIDEO_MODE
RANDOM_VIDEO_MODE	RANDOM_XVIDEO_MODE
RANDOM_FULLFRAME_VIDEO_MODE2	RANDOM_XVIDEO_MODE
SIM_VIDEO_MODE	NORMAL_XVIDEO_MODE
NORMAL_2N_MODE	NORMAL_XVIDEO_MODE
RANDOM_XC7500_2I_RESET_MODE	RANDOM_XVIDEO_MODE
RANDOM_XC_MODE	RANDOM_XVIDEO_MODE
RANDOM_XC_MODE_BOTH_CH	RANDOM_XVIDEO_MODE
RANDOM_XC_EX_MODE	RANDOM_EX_XVIDEO_MODE
RANDOM_XC_SIM_MODE	RANDOM_XVIDEO_MODE
RANDOM_XC_EX_SIM_MODE	RANDOM_EX_XVIDEO_MODE

Lib_get_video_input_mode

機　能

ビデオ入力モード参照

形　式

```
#include "f_video.h"
int Lib_get_video_input_mode ( int channel );
```

901	902	903	904	FVL/LNX	高分解能
○	△	○	○	○	

解　説

指定したチャネル番号のビデオ入力モードを参照します。

① ***channel*** はチャネル番号です。(0～3)

戻り値

入力モードを返します。

値	定　　数	意　　味
0x000000	NORMAL_VIDEO_MODE	通常入力モードに設定されています。
0x000001	STROBO_VIDEO_MODE	ストロボ入力モードに設定されています。
0x0000ff	RANDOM_VIDEO_MODE	ランダムトリガモードに設定されています。
0x0fffff	RANDOM_FULLFRAME_VIDEO_MODE2	ランダムトリガモード(フルフレーム)に設定されています。
0x001000	SIM_VIDEO_MODE	同時取込モードに設定されています。
0x001001	NORMAL_2N_MODE	2ライン取込ノンインターレースモードに設定されています。
0x001003	RANDOM_XC7500_2I_RESET_MODE	XC-7500 2ライン出力インターレースRESET モードに設定されています。
0x01000000	RANDOM_XC_MODE	XC55ランダムトリガモードに設定されています。
0x01000001	RANDOM_XC_MODE_BOTH_CH	XC55ランダムトリガ･2チャネル同時露光モードに設定されています。
0x01000002	RANDOM_XC_EX_MODE	XC55外部ランダムトリガモードに設定されています。
0x01000003	RANDOM_XC_SIM_MODE	XC55同時ランダムトリガモードに設定されています。
0x01000004	RANDOM_XC_EX_SIM_MODE	XC55外部・同時ランダムトリガモードに設定されています。

例

ビデオ入力モードを参照し、ランダムトリガモードでない場合は変更します。

```
#include "f_video.h"

void random_channel( int channel )
{
    if( NORMAL_VIDEO_MODE == Lib_get_video_input_mode ( channel ))
    {
        Lib_set_video_input_mode( RANDOM_VIDEO_MODE, channel );
    }
    Lib_video_channel( channel );
}
```

留意事項

○Lib_set_video_input_modeを参照してください。

902

○FVC02，FVC05は、対応しておりません。

FVL/LNX

○ビデオモードは、以下に示された3種類のいずれかとなります。

NORMAL_XVIDEO_MODE	:	通常入力モード
RANDOM_XVIDEO_MODE	:	ランダムトリガモード
RANDOM_EX_XVIDEO_MODE	:	外部ランダムトリガモード

Lib_set_xvideo_input_mode

機　能　　ビデオ入力モード設定2

形　式

```
#include "f_video.h"
int  Lib_set_xvideo_input_mode( int  iMode, int  iChannel, int  iCamera );
```

901	902	903	904	FVL/LNX	高分解能
	△			△	△

解　説

指定したチャネル番号のビデオ入力モードを設定します。FVC02,FVC05,FVC06,FV-GP440,FHC3312,及びFHC3313専用です。
※FVC05は、FVL/DOS, 902, FVL/LNXのみ対応
※FVC06は、FVL/LNXのみ対応
※FV-GP440は、FVL/LNXのみ対応
※FHC3312,FHC3313は、FVL/LNXのみ対応

① ***iMode*** は入力モードです。

値	定　　数	意　　味
0	NORMAL_XVIDEO_MODE	通常入力モードに設定します。
1	RANDOM_XVIDEO_MODE	内部ランダムトリガ入力モードに設定します。
2	RANDOM_EX_XVIDEO_MODE	外部ランダムトリガ入力モードに設定します。
3	ENCODER_XVIDEO_MODE	エンコーダを使用した取込モードに設定します。(FVC06, FHC3312, FHC3313専用)

② ***iChannel*** はチャネル番号です。

③ ***iCamera*** はカメラ種別です。
FVL/LNXでFVC06,FHC3312,FHC3313を使用する場合には、本パラメータは無視されます。

FVC02対応カメラ

定　　数	意　　味
XC_55	XC-55 ソニー製
CS_8530_01	CS8530-01 東芝テリー製
CS_8530D_01	CS8530D-01 東芝テリー製
CS_3720	CS3720 東芝テリー製
TI_480A	TI-480A 日興電気通信製
XC_HR300	XC-HR300　ソニー製
EIA_170_1	EIA-170仕様（TI-324系）
EIA_170_2	EIA-170仕様（TI-124系）
XC_HR50	XC-HR50 ソニー製
CS3730DC	CS3730DC東芝テリー製
XC_HR70	XC-HR70 ソニー製（※）
TI_1200A	TI-1200A 日興電気通信製（※）
CS8550D	CS8550D/CS8550Di東芝テリー製
CS8550I_01	CS8550i-01東芝テリー製

(※)FVL/DOS, 902ではサポートされていません。

FVC05対応カメラ（FVL/DOS, 902）

定　　数	意　　味
XC_55	XC-55 ソニー製
TI_480A	TI-480A 日興電気通信製
EIA_170_1	EIA-170仕様（TI-324系）
EIA_170_2	EIA-170仕様（TI-124系）
XC_HR50	XC-HR50 ソニー製
CS8550D	CS8550D/CS8550Di東芝テリー製
CS8560D	CS8560D/CS8570D東芝テリー製
KP_F30	KP-F30 日立国際電気製
KP_F33	KP-F33 日立国際電気製
IK_52V	IK-52V/IK-53V 東芝製

FVC05・FV-GP440対応カメラ(FVL/LNX)

定　　数	意　　味
XC_55	XC-55 ソニー製
TI_480A	TI-480A 日興電気通信製
TI_480A_640_480	TI-480A 日興電気通信製　640x480
EIA_170_1	EIA-170仕様（TI-324系）
EIA_170_2	EIA-170仕様（TI-124系）
EIA_170_2_640_480	EIA-170仕様（TI-124系）640x480
XC_HR50	XC-HR50 ソニー製
XC_HR50_640_480	XC-HR50 ソニー製 640x480
XC_HR70	XC-HR70 ソニー製 992x768
XC_HR70_512_480	XC-HR70 ソニー製 512x480
XC_HR70_1024_768	XC-HR70 ソニー製 1024x768
XC_HR90_P30fps_1280_960	XC-HR90 ソニー製 1280x960 30F（※1）
XC_HR90_P15fps_1280_960	XC-HR90 ソニー製 1280x960 15F（※1）
KP-F30_640_480	KP-F30 日立国際電気製 640x480
KP-F33_640_480	KP-F33 日立国際電気製 640x480
IK_52V_640_480	IK-52V/IK-53V 東芝製 640x480
CS8550D	CS8550D/CS8550Di東芝テリー製
CS8560D_640_480	CS8560D/CS8570D東芝テリー製 640x480
CS8550I_01	CS8550i-01東芝テリー製
CS8550I_01_640_480	CS8550i-01東芝テリー製 640x480
CS8550I_51	CS8550i-51東芝テリー製
CS8550I_51_640_480	CS8550i-51東芝テリー製 640x480
CS8550D_640_480	CS8550D/CS8550Di東芝テリー製 640x 480
CS8560D_P2_226	CS8560D 東芝テリー製 パーシャルスキャン 1/2
CS8560D_P4_80	CS8560D 東芝テリー製 パーシャルスキャン 1/4
CV_A1	CV-A1 JAI製 1376x1040
CV_A2	CV-A2 JAI製 1600x1200
STC_A33A_P60fps_640_480	STC-A33A 60F SENTECH製 640x480
STC_A33A_P60fps_P2_242	STC-A33A 60F SENTECH製 パーシャルスキャン 1/2
STC_A33A_P60fps_P4_104	STC-A33A 60F SENTECH製 パーシャルスキャン 1/4
STC_A33A_P90fps_640_480	STC-A33A 90F SENTECH製 640x480
STC_A33A_P90fps_P2_242	STC-A33A 90F SENTECH製 パーシャルスキャン 1/2
STC_A33A_P90fps_P4_104	STC-A33A 90F SENTECH製

	パーシャルスキャン 1/4
STC_A83A_1024_768	STC-A83A SENTECH製 1024x768
STC_A83A_P2_354	STC-A83A SENTECH製 パーシャルスキャン 1/2
STC_A83A_P4_146	STC-A83A SENTECH製 パーシャルスキャン 1/4
STC_A152A_1344_1040	STC-A152A SENTECH製
STC_A152A_P2_486	STC-A152A SENTECH製 パーシャルスキャン 1/2
STC_A152A_P4_198	STC-A152A SENTECH製 パーシャルスキャン 1/4
STC_A202A_1600_1200	STC-A202A SENTECH製 1600x1200
STC_A202A_P2_550	STC-A202A SENTECH製 パーシャルスキャン 1/2
STC_A202A_P4_211	STC-A202A SENTECH製 パーシャルスキャン 1/4

※1 FV-GP440 でのみ対応となります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

カメラを参照し内部ランダムトリガモードに設定します。

```
#include "f_video.h"

void set_xvideo_input_mode( int  iChannel )
{
int  iMode, iCamera

    if ( NORMAL_RETURN == Lib_get_xvideo_input_mode( &iMode,
                                                  iChannel, &iCamera ) )
{
        Lib_set_xvideo_input_mode( RANDOM_XVIDEO_MODE, iChannel, iCamera );
}
}
```

留意事項

共通

〇本関数は、FVC02およびFVC05専用です。FVC05は、FVL/DOS, 902, FVL/LNXのみ対応しています。
FVC06 は、FVL/LNX のみ対応しています。
FV-GP440 は、FVL/LNX のみ対応しています。
FHC3312, FHC3313 は、FVL/LNX のみ対応しています。
〇初期状態は、通常入力モードで、システム構築で設定されたカメラとなります。
〇FVC02, FVC05は、ボード内の2CHとも同じビデオモードが設定されます。
例）チャネル番号に0を指定するとCH0, CH1とも同じビデオモードとなります。
〇システム構築で設定したカメラは、Lib_get_xvideo_input_modeで取得可能です。
〇カメラ種別と実際に接続しているカメラは、必ず、一致させてください。
〇電源投入後の最初の呼び出し時は、画像取込を開始するまで200msのウェイトを入れてください。
〇EIA-170仕様のカメラを使用する場合は、必ず映像クロックを確認し、ご使用になるカメラに適した選択をしてください。

・EIA_170_1 …… TI-324系
・EIA_170_2 …… TI-124系

詳細は「FVC02取扱説明書」又は「FVC05取扱説明書」映像クロックの選択を参照して下さい。
〇FVC05でXC-55を使用する場合はオプションブラケットが必要です。詳しくは、高速アナログ画像入力ボード FVC05取扱説明書を参照下さい。

902

〇TI-1200AおよびXC_HR70は対応していません。
〇FVC05は、システム及びライブラリVer.4.20からの対応となります。それ以前のバージョンでは動作しません。
〇本ライブラリで設定するカメラ・モードが現状の設定と同じ場合は、何もせずに戻り値として-1を返します。

FVL/LNX

〇FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。
〇本関数を使用しカメラ種別を変更する場合に、取込サイズの違うカメラへの変更は行えません。
取込サイズの違うカメラへの変更を行う場合には、システムメニューから変更を行ってください。
〇TI-1200Aをトリガモードでご使用の場合、電子シャッタ速度設定に制限があります。
利用可能な電子シャッタ速度は、1/500、1/1000、1/2000、1/4000、1/10000のいずれかになります。
〇FVC05はVer.2.00からの対応となります。それ以前のバージョンでは動作しません。
〇FVC06 は Ver.2.10 からの対応となります。それ以前のバージョンでは動作しません。
〇FV-GP440 は Ver.3.00 からの対応となります。それ以前のバージョンでは動作しません。
〇FHC3312,FHC3313 は Ver.3.10 からの対応となります。それ以前のバージョンでは動作しません。
〇ENCODER_XVIDEO_MODEはVer.2.10からの対応となります。
それ以前のバージョンでは動作しません。

Lib_get_xvideo_input_mode

機　能

指定したチャネル番号のビデオ入力モードの参照

形　式

```
#include "f_video.h"
int  Lib_get_xvideo_input_mode( int  *iMode, int  iChannel, int  *iCamera );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	△

解　説

指定したチャネル番号のビデオ入力モードを参照します。

① ***iMode** は入力モードです。

値	定　　数	意　　味
0	NORMAL_XVIDEO_MODE	通常入力モードに設定されています。
1	RANDOM_XVIDEO_MODE	内部ランダムトリガ入力モードに設定されています。
2	RANDOM_EX_XVIDEO_MODE	外部ランダムトリガ入力モードに設定されています。
3	ENCODER_XVIDEO_MODE	エンコーダを使用した取込モードに設定します。(FVC06,FHC3312,FHC3313専用)

② ***iChannel*** はチャネル番号です。

③ *** iCamera** はカメラ種別です。

定　　数	意　　味
UNKNOWN	未選択
XC_55	XC-55 ソニー製
CS_8530_01	CS8530-01 東芝テリー製
CS_8530D_01	CS8530D-01 東芝テリー製
CS_3720	CS3720 東芝テリー製
TI_480A	TI-480A 日興電気通信製
XC_HR300	XC-HR300　ソニー製
EIA_170_1	EIA-170仕様（TI-324系）
EIA_170_2	EIA-170仕様（TI-124系）
XC_HR50	XC-HR50 ソニー製
CS3730DC	CS3730DC東芝テリー製
XC_HR70	XC-HR70 ソニー製（※）
TI_1200A	TI-1200A 日興電気通信製（※）
CS8550D	CS8550D/CS8550Di東芝テリー製
CS8550I_01	CS8550i-01東芝テリー製
CS8560D	CS8560D/CS8570D東芝テリー製
KP_F30	KP-F30 日立国際電気製
KP_F33	KP-F33 日立国際電気製
IK_52V	IK-52V/IK-53V 東芝製
CS8560D_640_480	CS8560D/CS8570D東芝テリー製　640x480
KP-F30_640_480	KP-F30 日立国際電気製 640x480
KP-F33_640_480	KP-F33 日立国際電気製 640x480
IK_52V_640_480	IK-52V/IK-53V 東芝製　640x480
TI_480A_640_480	TI-480A 日興電気通信製　640x480

EIA_170_2_640_480	EIA-170仕様（TI-124系）640x480
XC_HR50_640_480	XC-HR50 ソニー製 640x480
CS8550D_640_480	CS8550D/CS8550Di東芝テリー製　640x 480
CS8560D_P2_226	CS8560D1/CS8570D 東芝テリー製 パーシャルスキャン 1/2
CS8560D_P4_80	CS8560D1/CS8570D 東芝テリー製 パーシャルスキャン 1/4
CS8550I_01	CS8550i-01東芝テリー製
CS8550I_01_640_480	CS8550i-01東芝テリー製 640x480
CS8550I_51	CS8550i-51東芝テリー製
CS8550I_51_640_480	CS8550i-51東芝テリー製 640x480
XC_HR70	XC-HR70 ソニー製 992x768
XC_HR70_512_480	XC-HR70 ソニー製 512x480
XC_HR70_1024_768	XC-HR70 ソニー製 1024x768
CV_A1	CV-A1 JAI製 1376x1040
CV_A2	CV-A2 JAI製 1600x1200
STC_A33A_P60fps_640_480	STC-A33A 60F SENTECH製 640x480
STC_A33A_P60fps_P2_242	STC-A33A 60F SENTECH製 パーシャルスキャン 1/2
STC_A33A_P60fps_P4_104	STC-A33A 60F SENTECH製 パーシャルスキャン 1/4
STC_A33A_P90fps_640_480	STC-A33A 90F SENTECH製 640x480
STC_A33A_P90fps_P2_242	STC-A33A 90F SENTECH製 パーシャルスキャン 1/2
STC_A33A_P90fps_P4_104	STC-A33A 90F SENTECH製 パーシャルスキャン 1/4
STC_A83A_1024_768	STC-A83A SENTECH製 1024x768
STC_A83A_P2_354	STC-A83A SENTECH製 パーシャルスキャン 1/2
STC_A83A_P4_146	STC-A83A SENTECH製 パーシャルスキャン 1/4
STC_A152A_1344_1040	STC-A152A SENTECH製
STC_A152A_P2_486	STC-A152A SENTECH製 パーシャルスキャン 1/2
STC_A152A_P4_198	STC-A152A SENTECH製 パーシャルスキャン 1/4
STC_A202A_1600_1200	STC-A202A SENTECH製 1600x1200
STC_A202A_P2_550	STC-A202A SENTECH製 パーシャルスキャン 1/2
STC_A202A_P4_211	STC-A202A SENTECH製 パーシャルスキャン 1/4

戻り値

処理結果

値	定　　数	意　　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

Lib_get_xvideo_input_mode の例を参照してください。

留意事項

○本関数は、FVC02,FVC05,FVC06,FV-GP440,FHC3312及びFHC3313専用です。
※FVC06はFVL/LNXでのみ対応です。
※FV-GP440はFVL/LNXでのみ対応です。
※FHC3312,FHC3313はFVL/LNXでのみ対応です。

○FVC02, FVC05は、ボード内の2CHとも同じビデオモードが設定されます。
　例）チャネル番号に 0 を指定するとCH0, CH1とも同じビデオモードとなります。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_channel

機　能

ビデオ入力チャネル、オフセット、2値レベル設定

形　式

```
#include "f_video.h"
int Lib_channel( int flag, int channel, int offset, int bin_level );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

カレントビデオ入力チャネルを変更し、同時にオフセットレベル、2値化レベルを変更します。
フラグにより、チャネル切り換えの完了まで待ちます。
チャネル切り換えは、同期信号の安定待ちのための時間が2フレーム時間確保されており、切り換え指令から数フレーム時間経過して始めて切り換えが完了した事、としています。

① ***flag*** はチャネル切り換えウエイトフラグです。

値	定　数	意　味
0	NO_WAIT	設定後、切り換えの完了を待たずに終了します。
1	WAITING	切り換えの完了を確認後に、終了します。

② ***channel*** はチャネル番号(0～3)です。
901AT，903，FVC01の1枚搭載機種では0～1となります。

③ ***offset*** はオフセットレベル(-128～127)です。

④ ***bin_level*** は2値化レベル(0～255)です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル0(オフセット=0, 2値化レベル=128)から画像を入力します。

```
#include "f_video.h"

#define  SET_CHANNEL  0
#define  SET_BINARY   128
#define  SET_OFFSET   0

void set_channel()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_freerun( );
}
```

留意事項

共通

○システムパラメータの「カレントチャネル」、「オフセットレベル」、「2値化レベル」が指定したパラメータにより、更新されます。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。
○チャネル切り替えウエイトフラグをNO_WAITで使用する場合は、切替えるチャネル間でのカメラ同期を取る必要があります。

901

○切り替え指示から2フレーム経過して始めて切り替えが完了します。

902

○VCAP、CATSは、切り替え指示から2フレーム経過して始めて切り替えが完了します。
RICE は、切り替え指示から1フレーム経過して始めて切り替えが完了します。
FVC01, FVC02, FVC05は、切り替えの完了をまたずに終了します。

903

○切り替え指示から1フレーム経過して始めて切り替えが完了します。

904

○切り替え指示から1フレーム経過して始めて切り替えが完了します。

FVL/LNX

○VCAP, CATSは切り替え指示から2フレーム経過して始めて切り替えが完了します。
RICEは、切り替え指示から1フレーム経過して始めて切り替えが完了します。
FVC01, FVC02, FVC05, FVC06, FV-GP440, FHC3312, FHC3313は、切り替えの完了をまたずに終了します。

Lib_video_channel

機　能

ビデオ入力チャネル設定

形　式

```
#include "f_video.h"
int Lib_video_channel( int channel );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

カレントビデオ入力チャネルを変更します。切り換えの完了は待ちません。

① ***channel*** はチャネル番号です。
901AT，903，FVC01，FVC02，FVC05，FVC06，FHC3312 の1枚搭載機種では0～1となります。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル1から画像を入力します。

```
#include "f_video.h"

#define  SET_CHANNEL  1

void set_channel()
{
    Lib_video_channel( SET_CHANNEL );
    Lib_freerun( );
}
```

留意事項

○システムパラメータの「カレントチャネル」が指定したパラメータにより、更新されます。
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。
○本関数でチャネルを切替えるには、切替えるチャネル間でのカメラ同期を取る必要があります。

Lib_xvideo_channel

機　能

ビデオ入力チャネル設定Ⅱ

形　式

```
#include "f_video.h"
  int Lib_xvideo_channel( int wait_opt, int channel );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

カレントビデオ入力チャネルを変更します。
切り換えの完了はパラメータにより行います。

① ***wait_opt*** はチャネルの切り換えオプションです。

値	定　　数	意　　味
0	NO_WAIT	設定後、切り換えの完了を待たずに終了します。
1	WAITING	切り換えの完了を確認後に、終了します。

② ***channel*** はチャネル番号です。
901AT, 903, FVC01, FVC02, FVC05, FVC06, FHC3312 の1枚搭載機種では0～1となります。FV-GP440の1枚搭載機種では0～3となります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル1から画像を入力します。

```
#include "f_video.h"

#define   SET_CHANNEL   1

void set_channel()
{
    Lib_xvideo_channel( WAITING, SET_CHANNEL );

    Lib_freerun( );
}
```

留意事項

○システムパラメータの「カレントチャネル」が指定したパラメータにより、更新されます。
○FVC01, FVC02, FVC05, FVC06, 904(a/ac/p/pc), FV-GP440, FHC3312, FHC3313は、WAITING を指定しても切り替えの完了をまたずに終了します。
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。
○チャネル切り替えオプションをNO_WAITで使用する場合は、切替えるチャネル間でのカメラ同期を取る必要があります。

Lib_random_video_channel

機　能

ランダムトリガビデオ入力チャネル設定

形　式

```
#include "f_video.h"
int Lib_random_video_channel( int channel );
```

901	902	903	904	FVL/LNX	高分解能
○			△		

解　説

外部入出力端子を使用したランダムトリガビデオ入力チャネルに変更します。

① ***channel*** はチャネル番号(0～3)です

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル1から画像を入力します。

```
#include "f_video.h"

#define  SET_CHANNEL  1

void set_channel()
{
    Lib_random_video_channel( SET_CHANNEL );

    Lib_freerun( );
}
```

留意事項

○通常のビデオ入力に戻る場合は前項の Lib_set_video_input_mode を実行します。

○TI-324Aの設定方法を以下のような設定にする必要があります。
① インターレースモード
② フィールド蓄積モード
③ SYNCリセットモード
シャッタ露出時間によって映像の取込時間が変化します。
※詳細は「TI-324A説明書」を参照してください。

◎ランダムトリガモードでの映像入力時間
Ts ：シャッタ露出時間
Taq：ランダムトリガ出力から映像入力完了までの所要時間

Ts(s)	Taq (ms)
1/60(OFF)	33.3
1/100	26.2
1/250	20.3
1/500	18.3
1/1000	17.2
1/2000	16.7
1/4000	16.7
1/10000	16.7

◎外部入力端子(各4点ずつ)は、TI-324Aとのコントロール信号として使用されます。
接続方法等の詳細は各「取扱説明書」を参照してください。

◎画像メモリへの入力ビデオの取込形式
入力ビデオは画像メモリの上半分に取込まれます。

○本ライブラリはFV904p, FV904pcではサポートしておりません。

Lib_video_offset

機　能

ビデオ・オフセットレベル設定

形　式

```
#include "f_video.h"
int Lib_video_offset( int offset );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	

解　説

カレントビデオ入力チャネルのビデオ入力オフセットレベルを指定します。

① ***offset*** はオフセットレベル(-128～127)です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

オフセットレベルを5に設定します。

```
#include "f_video.h"

#define  SET_OFFSET  5

void set_offset()
{
    Lib_video_offset( SET_OFFSET );

    Lib_freerun( );
}
```

留意事項

共通

○システムパラメータの「カレントチャネル」の「オフセットレベル」が指定したパラメータにより、更新されます。

901, 904

○実際に更新されるのはFT(フレームトップ)になります。

Lib_video_bin_level

機　能

ビデオ2値化レベル設定

形　式

```
#include "f_video.h"
int Lib_video_bin_level( int bin_level );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	

解　説

カレントビデオ入力チャネルのビデオ2値化レベル制御方式を指定します。

① ***bin_level*** は2値化レベル（0～255）です。

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

戻り値

処理結果

例

2値化レベルを127に設定します。

```
#include "f_video.h"

#define  SET_BINLEVEL  127
void set_binlevel()
{
    Lib_video_bin_level( SET_BINLEVEL );

    Lib_freerun( );
}
```

留意事項

共通

○システムパラメータの「カレントチャネル」の「2 値化レベル」が指定したパラメータにより、更新されます。

901, 904

○実際に更新されるのは FT（フレームトップ）になります。

Lib_video_transmit

機　能

フレームバッファへの画像転送

形　式

```
#include "f_video.h"
int Lib_video_transmit( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

カレントメモリ上の画像データをフレームバッファへ転送します。

① *flag* は転送メモリです。
　(該当するメモリのビットをオンにしたビットパタンで指定します)

値	定　　数	意　　味
0x01	BIN_PLANE	カレント2値メモリ番号の内容を転送します。
0x20	GRAY_PLANE	カレント濃淡メモリ番号の内容を転送します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオチャネル0(オフセット=0, 2値化レベル=128)から画像を入力し、前処理後、画像を表示します。

```
#include "f_video.h"

#define  SET_CHANNEL   0
#define  SET_BINARY    128
#define  SET_OFFSET    0

void set_transmit()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE|GRAY_PLANE );

    Lib_freeze( NOT_TRANSMIT );

    /* 前処理 */

    Lib_video_transmit( BIN_PLANE|GRAY_PLANE );
}
```

留意事項

〇画像の転送
画像の転送を行わない場合、表示画面(フレームバッファ)とメモリ内容は違います。
すなわち画像変形処理を行った場合、その画像は改めて転送しないと表示されません。

〇複数指定の場合
2値のメモリを転送後、濃淡のメモリを転送します。

〇高分解能カメラを使用する場合は、Lib_ex_video_transmitまたは
Lib_lsensor_video_transmitを使用してください。
(Lib_lsensor_video_transmitに関しては、「902高分解能センサライブラリ説明書」を
ご参照下さい。)

〇FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_xvideo_transmit

機　能

フレームバッファへの画像転送Ⅱ

形　式

```
#include "f_video.h"
int  Lib_xvideo_transmit( int memory_no, int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

指定メモリ上の画像データをフレームバッファへ転送します。

① ***memory_no*** はメモリ番号です。

② ***flag*** は転送メモリです。
（該当するメモリのビットをオンにしたビットパタンで指定します）

値	定　　数	意　　味
0x01	BIN_PLANE	指定された2値メモリ番号の内容を転送します。
0x20	GRAY_PLANE	指定された濃淡メモリ番号の内容を転送します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

メモリ番号1にビデオチャネル0(オフセット=0, 2値化レベル=128)から画像を入力します。

```
#include "f_stdLib.h"
#include "f_video.h"

#define  SET_CHANNEL   0
#define  SET_BINARY    128
#define  SET_OFFSET    0
#define  SET_MEMORY_NO 1

void set_transmit()
{
    Lib_change_memory ( SET_MEMORY_NO );

    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE|GRAY_PLANE );

    Lib_freeze( NOT_TRANSMIT );
    /* 処理 */
    Lib_xvideo_transmit(SET_MEMORY_NO,(BIN_PLANE|GRAY_PLANE));
}
```

留意事項

○複数指定の場合
　2値のメモリを転送後、濃淡のメモリを転送します。
○実行前に、Lib_freezeを必ず行ってください。
○高分解能カメラを使用する場合は、Lib_ex_video_transmit
　または、Lib_lsensor_video_transmitを使用してください。
　(Lib_lsensor_video_transmitに関しては、「902高分解能センサライブラリ説明書」を
　ご参照下さい。)
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_stage_input

機　能

処理ステージパラメータ適用

形　式

```
#include "f_video.h"
void Lib_stage_input( void );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

カレントステージNo.の下記パラメータを参照し、その値を適用します。
処理範囲としてウィンドウを表示します。
☆入力ファイル(ビデオ入力チャネル)
☆処理ファイル(入力メモリ番号)
☆処理範囲(ウィンドウサイズ)

戻り値

ありません。

例

画像を入力後、何らかの処理を行います。

```
#include "f_video.h"
#include "f_pinf.h"

void set_video()
{
    int xs, ys, xe, ye;

    Lib_stage_input();

    Lib_get_stage_window( &xs, &ys, &xe, &ye );  /* 処理範囲の参照 */
    /* 処理 */
}
```

留意事項

○ステージNo.を指定した画像入力を行う場合は次項のLib_set_stage_inputを使用してください。
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_set_stage_input

機　能

処理ステージパラメータ適用Ⅱ

形　式

```
#include "f_video.h"
void Lib_set_stage_input( int stage_no );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

指定ステージNo.の下記パラメータを参照し、その値を適用します。
　処理範囲としてウィンドウを表示します。
　　☆入力ファイル(ビデオ入力チャネル)
　　☆処理ファイル(入力メモリ番号)
　　☆処理範囲（ウィンドウサイズ）

① ***stage_no*** はステージパラメータNo.(0～127)です。

戻り値

ありません。

例

画像を入力後、何らかの処理を行います。

```
#include "f_video.h"
#include "f_pinf.h"
#define  SET_STAGE_NO  10

void set_video()
{
    int xs, ys, xe, ye;

    Lib_set_stage_input( SET_STAGE_NO );

    Lib_get_stage_window( &xs, &ys, &xe, &ye );  /* 処理範囲の参照 */
    /* 処理 */
}
```

留意事項

○システムパラメータの「カレントステージNo.」が指定したパラメータにより、更新されます。
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_input_video_control

機　能

入力ビデオ制御

形　式

```
#include "f_video.h"
void Lib_input_video_control( int ctrl );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	○

解　説

メモリに取り込むビデオの選択を行います。

① *ctrl* は入力ビデオです。
（該当するメモリのビットをオンにしたビットパタンで指定します。）

値	定　　数	意　　味
0x01	BIN_PLANE	2値画像を選択します。
0x20	GRAY_PLANE	濃淡画像を選択します。

戻り値

ありません。

例

ビデオチャネル0(オフセット=0, 2値化レベル=128)から2値画像の入力をします。

```
#include "f_video.h"

#define  SET_CHANNEL  0
#define  SET_BINARY   128
#define  SET_OFFSET   0

void set_freerun()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( BIN_PLANE );

    Lib_freerun( );
}
```

留意事項

○モニタ表示の制御は次項のLib_display_controlで行います。

○システムパラメータの「入力ビデオ制御」が、指定したパラメータにより更新されます。

Lib_display_control

機　能

モニタTV表示項目制御＆描画制御

形　式

```
#include "f_video.h"
int Lib_display_control( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	○

解　説

モニタTVの表示項目、描画ライブラリの動作を制御します。

① *flag* は表示フレームバッファ、描画ライブラリ動作指定フラグです。
（以下のオプションをOR指定します。）

値	定　数	意　味
0x00000001	BIN_PLANE	2値用フレームバッファを表示します。
0x00000020	GRAY_PLANE	濃淡用フレームバッファを表示します。
0x00000040	LINE_PLANE	線画用フレームバッファを表示します。
0x00000080	CHAR_PLANE	文字用フレームバッファを表示します。
FVL/LNXをお使いの場合		
0x40000000	VOUT_NOT_WAIT	描画ライブラリ実行時に、描画完了を待機しません。(描画指示は発行します)
0x80000000	VOUT_NOT_SYNC	描画ライブラリ実行時に、描画指示を発行しません。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例1

モニタTV表示項目を2値、線画、文字を選択します。

```
#include "f_video.h"

void set_freerun()
{

    Lib_input_video_control( BIN_PLANE );

    Lib_display_control( BIN_PLANE|LINE_PLANE|CHAR_PLANE );

    Lib_freerun( );
}
```

FVL/LNX

例2

モニタTV表示項目に濃淡、線画、文字を選択し、文字を描画完了待ちなしで描画します。

```
void func( void )
{
        Lib_display_control( GRAY_PLANE|LINE_PLANE|CHAR_PLANE|VOUT_NOT_WAIT );
        Lib_chrdisp( 1, 1, "ABCDE" );
        Lib_display_keyinput( 0, 450, "OK" );
}
```

留意事項

共通

〇システムパラメータの「モニタTV表示項目」が指定したパラメータにより、更新されます。

FVL/LNX

〇VOUT_NOT_WAITを指定した場合、実際の描画はアイドル時(CPUを使っていないとき)に行なわれます。上記の例ではキー入力待機時に描画が行なわれます。
描画ライブラリを連続してコールするような場合にはあまり効果がありませんので、ご注意下さい。

〇VOUT_NOT_SYNCを指定した場合、実際の描画は“Lib_vout_sync_framebuffer()”関数コール時に行なわれます。この場合のプログラム例は、「90X基本ライブラリ説明書 Vol.2」のLib_vout_sync_framebufferの項を参照してください。
また、VOUT_NOT_WAITと組み合わせて使用することも可能です。

■Lib_vout_sync_framebuffer関数以外の再描画のタイミングについて

FVL/LNXで使用しているwindowでは、windowの再描画が必要な場合に、描画イベントを発行しています。
例えば、あるパッドによってそのwindowの一部が隠されてしまい、パッド移動後に再度window内部の表示が必要になった場合には、windowの再描画のイベント命令を発行します。これは、windowのサイズを変更した場合や、windowの一部を画面外等に隠してしまった場合にも同様です。

そのような動作を行った場合には自動的に描画イベントが発行されて、Lib_display_control関数にてVOUT_NOT_SYNCオプションを付加していても、window内部が再描画されてしまいます。

■処理の内部でsync_framebufferを呼び出している関数
以下の関数は、処理内部でLib_vout_sync_framebufferを呼び出しています。

- Lib_gs_search(displayオプションがonの時)
- Lib_ascope
- Lib_histogram

そのため、上記関数を実行した場合には、VOUT_NOT_SYNC を指定していても、描画が実行されてしまいます。

Lib_gray_scale

機　能

グレイスケール制御

形　式

```
#include "f_video.h"
void Lib_gray_scale( int flag );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○		○

解　説

モニタTV表示に関する、グレイスケールの表示制御を行います。

① ***flag*** は表示オプションです

値	定　数	意　味
0x00	OFF	グレイスケールを表示しません。
0x01	ON	グレイスケールを表示します。

戻り値

ありません。

例

グレイスケールを表示します。

```
#include "f_video.h"

void set_scale()
{
    Lib_gray_scale(ON);

    Lib_freerun( );
}
```

留意事項

○システムパラメータの「モニタTV表示オプション」が、指定されたパラメータにより更新されます。

Lib_window

機　能

処理ウィンドウの設定

形　式

```
#include "f_video.h"
int Lib_window( int xs, int ys, int xe, int ye );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

現在のビデオ入力チャネル側の処理ウィンドウとモニタTVの表示範囲を指定します。
ただし、文字・線画の表示は、この｢処理ウィンドウ｣による制約を受けませんので、｢絵｣の映る部分と、｢文字｣の映る部分を分離した画像を表示する事もできます。

① *xs* はX位置始点です。

② *ys* はY位置始点です。

③ *xe* はX位置終点です。

④ *ye* はY位置終点です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

(10, 10) - (400, 400)のウィンドウを表示します。

```
#include "f_video.h"
#define XS_POS   10
#define YS_POS   10
#define XE_POS   400
#define YE_POS   400

void set_window()
{
    Lib_window( XS_POS, YS_POS, XE_POS, YE_POS );

    Lib_freerun( );
}
```

留意事項

共通

○システムパラメータの(カレントステージNo.)「ウィンドウ範囲」が指定したパラメータにより、
　更新されます。

○画像メモリの処理範囲だけを設定する場合は、Lib_set_stage_window を使用してください。
　(Lib_set_stage_window に関しては、「90X基本ライブラリ説明書 Vol.3」をご参照下さい。)

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

901

○本ライブラリはフレームバッファの制御を行っているため数10msの処理時間を要します。

902

○本ライブラリはフレームバッファの制御を行っているため以下の処理時間を要します。

VCAP　　　：86ms
CATS　　　：26ms
RICE, FVC01：12ms

○高分解能カメラを使用する場合は Lib_set_stage_window を使用してください。
　(Lib_set_stage_window に関しては、「90X基本ライブラリ説明書 Vol.3」をご参照下さい。)

903

○本ライブラリはフレームバッファの制御を行っているため数10msの処理時間を要します。

904

○本ライブラリはフレームバッファの制御を行っているため数10msの処理時間を要します。

Lib_framework

機　能

処理ウィンドウの設定(枠表示版)

形　式

```
#include "f_video.h"
int Lib_framework( int xs, int ys, int xe, int ye );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

現在のビデオ入力チャネル側の処理ウィンドウとモニタTVの表示範囲を指定します。
ただし、枠線表示のみですので、この「処理ウィンドウ」範囲外の画像も表示されてしまいます。

① *xs* はX位置始点です。

② *ys* はY位置始点です。

③ *xe* はX位置終点です。

④ *ye* はY位置終点です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

(10, 10) - (400, 400)のウィンドウ枠を表示します。

```
#include "f_video.h"
#define XS_POS   10
#define YS_POS   10
#define XE_POS   400
#define YE_POS   400

void set_window()
{
    Lib_framework( XS_POS, YS_POS, XE_POS, YE_POS );
    Lib_freerun( );
}
```

留意事項

○システムパラメータの(カレントステージNo.)「ウィンドウ範囲」が、指定したパラメータにより更新されます。
○画像メモリの処理範囲だけを設定する場合は、Lib_set_stage_windowを使用してください。(Lib_set_stage_window に関しては、「90X基本ライブラリ説明書 Vol.3」をご参照下さい。)
○高分解能カメラを使用する場合はLib_set_stage_windowを使用してください。
○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_get_frame_ratio

機　能　　拡張画像メモリからフレームバッファへの縮尺参照

形　式

```
#include "f_video.h"
int Lib_get_frame_ratio( void );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	○

解　説

m×n時のメモリからフレームバッファへの縮尺を取得します。
フレーム・メモリへの縮小転送は、XY方向の長い方に合わせ小数点以下は切り上げています。

例えば横方向が素数が2048の場合、フレームバッファの画素数が512ですから
2048÷512=4が返ります。
〈例〉　　　2048×480の場合は 4が返ります。
　　　　　　512×2048の場合は 5が返ります。

戻り値　　縮小率

例

2値計測を行い、縮小された画像に計測結果を表示します。

```
#include "f_video.h"
#include "f_pinf.h"
#include "f_meas.h"

static struct MEASDATA *measp;    /* 1次特徴量テーブル */
static struct CALCDATA *calcultp; /* 2次特徴量テーブル */
static    int          indextbl[MAXBLOB+1];

void bin_execute()
{
        int scale;

        scale = Lib_get_frame_ratio();
        if( NORMAL_RETURN == Lib_measure( Lib_get_measure_no(), &measp ))
        {
             if ( OFF != Lib_orderng( Lib_get_measure_no(),
                                                       measp, indextbl ) )
             {
                    if(NORMAL_RETURN == Lib_calcult( 1, indextbl[0],
                                                              &calcultp ))
                    {
                             Lib_calcult_disp_result
                              ( 0, 0, scale, scale, indextbl[0],
                                                       1, measp, calcultp );
                    }
              }
        }
}
```

留意事項

ありません。

Lib_extraction_display_memory

機　能　　拡張画像メモリの通常サイズ表示

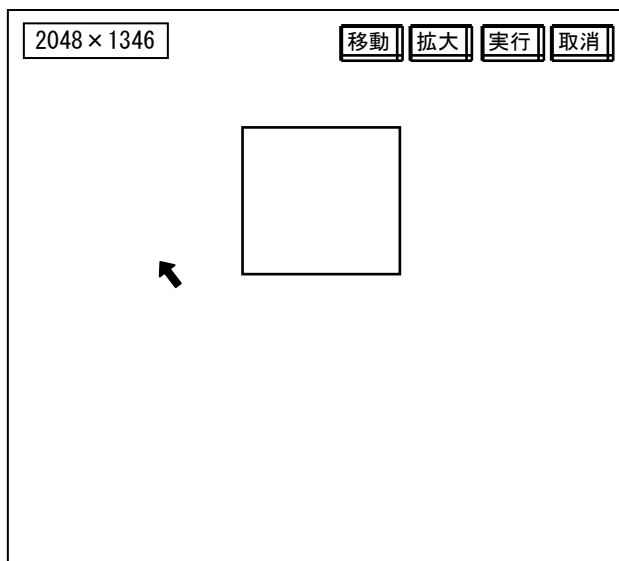
形　式

```
#include "f_video.h"
int  Lib_extraction_display_memory( int memory_no );
```

901	902	903	904	FVL/LNX	高分解能
○	○				○

解　説

指定したメモリ番号の画像をモニタTV上にメモリ全体が表示するように縮小表示し、モニタTV上で、512×480の範囲を選択し、通常表示します。

① ***memory_no*** はメモリ番号です。(0～)

○移動(MOVE)
　表示用のウィンドウを移動します。移動を中止する時は実行キーを押してください。

○拡大(ZOOM)
　表示用のウィンドウ部分を通常表示に拡大します。
　拡大時もトラックボールにより移動できます。中止する時は実行キーを押してください。

○実行(EXEC)
　ウィンドウ位置を登録し、終了します。

○取消(CANC)
　終了します。

※( )は英語モード時の表示内容

戻り値

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

初期化後、切り出しウィンドウを実行します。

```
#include "f_video.h"
#include "f_lsens.h"

void init()
{
        Lib_init_cursor();
        if( NORMAL_RETURN  == Lib_lsensor_init
                    ( NORMAL_FREEZE, D2048CHN, 2400, 0, 2048, 1024 ))
        {
                Lib_lsensor_video_input();
                Lib_extraction_display_memory( 0 );
        }
}
```

留意事項

○Lib_init_cursorを実行前にコールする必要があります。
(Lib_init_cursor に関しては、「90X基本ライブラリ説明書 Vol.2」をご参照下さい。)

○この関数を実行する前にメモリ上に画像入力しておく必要があります。

○Lib_input_video_controlで指定されたプレーン(2値または濃淡のどちらかのみ)を表示します。

Lib_ex_video_transmit

機　能

拡張画像メモリ時用のフレームバッファへの転送

形　式

```
#include "f_video.h"
int Lib_ex_video_transmit( int memory_no, int video_control, int mode );
```

901	902	903	904	FVL/LNX	高分解能
○	○			○	○

解　説

m×nの拡張画像メモリをフレームバッファへ転送します。

① ***memory_no*** はメモリ番号です。

② ***video_control*** はメモリの種類です。
（該当するメモリのビットをオンにしたビットパタンで指定します。）

値	定　　数	意　　味
0x20	GRAY_PLANE	濃淡のメモリを選択します。
0x01	BIN_PLANE	2値のメモリを選択します。

③ ***mode*** は転送モードです。

値	定　　数	意　　味
0	EXTRACT_MODE	切り出し転送を行います。
1	REDUCE_MODE	縮小転送を行います。

切り出し転送時には、Lib_get_extract_windowを参照し、その範囲を転送します。
(Lib_get_extract_window に関しては、「90X基本ライブラリ説明書 Vol.3」をご参照下さい。)

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

画像メモリ範囲の左上側をモニタTVに表示します。

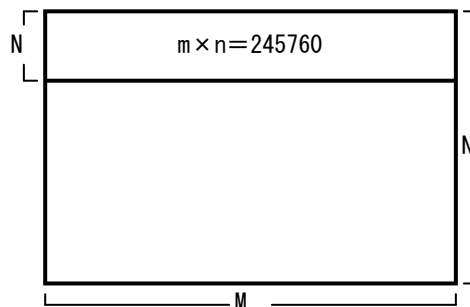
```
#include "f_stdLib.h"
#include "f_graph.h"
#include "f_pinf.h"

void transmit( )
{
     int xs, xe;
     int ys, ye;

     xs = 0;
     xe = Lib_get_dx_size() - 1;
     ys = 0;
     ye = Lib_get_dy_size() - 1;

     Lib_set_extract_window( xs, ys, xe, ye );
     Lib_ex_video_transmit( CURRENT_MEMORY, GRAY_PLANE, EXTRACT_MODE )
}
```

留意事項

○ Lib_video_transmitは、メモリの先頭アドレスから512×480バイトを転送するだけです。

245760バイトの連続した領域が転送されます。
(245760 = 512×480)

○ 複数指定の場合
2値のメモリを転送後、濃淡のメモリを転送します。

Lib_ex_disp_gray_memory

機　能

ワーク・メモリの濃淡画像用フレームバッファへの出力

形　式

```
#include "f_video.h"
int Lib_ex_disp_gray_memory ( void );
```

901	902	903	904	FVL/LNX	高分解能
○	○				○

解　説

拡張画像メモリ時にワーク・メモリの内容を濃淡画像用フレームバッファへ転送し、モニタTVに表示します。
転送サイズは512×480バイトです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

Lib_get_adrs_work_memory 参照してください。
（Lib_get_adrs_work_memory に関しては、「90X基本ライブラリ説明書 Vol.2」をご参照下さい。）

留意事項

○m×nの拡張画像メモリ(ラインセンサなど)モード以外の時にコールした場合は
ERROR_RETURN が返ります。

Lib_ex_disp_bin_memory

機　能

ワーク・メモリの2値画像用フレームバッファへの出力

形　式

```
#include "f_video.h"
int Lib_ex_disp_bin_memory ( void );
```

901	902	903	904	FVL/LNX	高分解能
○	○				○

解　説

拡張画像メモリ時にワーク・メモリの内容を2値画像用フレームバッファへ転送し、
モニタTVに表示します。
転送サイズは512×480÷8バイトです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

2値拡張画像メモリをワーク・メモリに転送し、モニタTVに表示します。

```
#include "f_stdLib.h"
#include "f_video.h"

void bin_memory_move( int memory_no )
{
        unsigned int *lmem_top_adrs, *mem_top_adrs, *lwork_adrs, *work_adrs;
        unsigned int y_offset,ly_offset;        /*Y方向オフセットサイズ*/
        int fx_size, fy_size;                   /*メモリのサイズ*/
        int dx_size, dy_size;                   /*フレームバッファのサイズ*/
        int xloop, x, y;

        /*フレームバッファ・サイズの取得*/
        dx_size = Lib_get_dx_size();  dy_size = Lib_get_dy_size();
        /*メモリ・サイズの取得*/
        fx_size = Lib_get_fx_size();  fy_size = Lib_get_fy_size();

        xloop = dx_size / LONG_ACCESS;
        ly_offset = fx_size / LONG_ACCESS;
        y_offset  = dx_size / LONG_ACCESS;

        /*画像メモリのアドレス取得*/
        if( (unsignd int *)NULL != (lmem_top_adrs = (unsigned int *)
                                Lib_adrs_bin_memory ( memory_no )) )
        {
                /*フレームバッファ転送用メモリのアドレス取得*/
                mem_top_adrs  = Lib_get_adrs_work_memory( );

                for( y = 0; y < dy_size; y++ )
                {
                        lwork_adrs = lmem_top_adrs;
                        work_adrs  = mem_top_adrs;
                        for( x = xs; x < x_loop ; x++ )
                                *work_adrs++ = *lwork_adrs++;
                        lmem_top_adrs += ly_offset;
                        mem_top_adrs  += y_offset;
                }
                Lib_ex_disp_bin_memory();
        }
}
```

留意事項

○m×nの拡張画像メモリ(ラインセンサなど)モード以外の時にコールした場合は
ERROR_RETURN が返ります。

Lib_frame_chain_freeze

機　能

連続フレームビデオフリーズ(カメラ切り替えなし)

形　式

```
#include "f_video.h"
int Lib_frame_chain_freeze( int start_memory_no, int frame_su );
```

901	902	903	904	FVL/LNX	高分解能
△	△	○	○	○	△

解　説

カレントビデオ入力チャネルの連続画像を開始メモリ以降、取得フレーム数分取込ます。

① ***start_memory_no*** は開始メモリ番号です。

② ***frame_su*** は取得フレーム数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

2フレーム分の連続画像をグレイメモリ0, 1へ取込ます。

```
#include "f_video.h"

#define START_MEMORY_NO     0
#define FRAME_SU            2

void input_video()
{
    Lib_input_video_control( GRAY_PLANE );

    Lib_set_field_squence( SET_ANY_FIELD_CONTINUOUS );

    Lib_frame_chain_freeze( START_MEMORY_NO, FRAME_SU );

    Lib_set_field_squence( SET_ANY_FIELD );
}
```

留意事項

共通

○開始メモリ以降、取得フレーム数分の画像メモリは事前に確保しておいてください。

○ビデオ入力モードは通常入力モードでのみ動作します。

○取得フレーム数が17以上の場合は、Lib_xframe_chain_freezeを指定してください。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

901

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：（ Lib_get_fx_size() + 16 ）＊ Lib_get_fy_size() ＊ 取得フレーム数
2値画像 ：（ Lib_get_fx_size() / 8 + 16 ）＊ Lib_get_fy_size() ＊ 取得フレーム数

○事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

○Lib_input_video_controlにより、取り込む入力映像を指定できます。

○本ライブラリは、CSC901, CSC901a, CSC901SBではサポートしておりません。

902

○RICE, FVC01 は2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：16 + 16 ＊ Lib_get_fy_size() ＊ 取得フレーム数

○事前にLib_set_field_squence にてフィールド・シーケンスを任意のフィールド入力に設定しておいてください。
RICEでは、Lib_set_field_squence(SET_ANY_FIELD)で御使用下さい。

○本ライブラリはVCAP, CATSではサポートしておりません。

903

○2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：(Lib_get_fx_size() + 16 ）＊ Lib_get_fy_size() ＊ 取得フレーム数

904

○2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：（ Lib_get_fx_size() + 16 ）＊ Lib_get_fy_size() ＊ 取得フレーム数

○事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
16 * Lib_get_fy_size() * 取得フレーム数

○Lib_input_video_control により、取り込む入力映像を指定できます。

○FVC06,FHC3312, FHC3313 をランダムトリガモードで使用する場合には、1 フレーム辺りの取り込み時間が Lib_capt_set_reg 関数の FVC06_TRIGGER_CYCLETEIME(FVC06 の場合)パラメータで指定した値になります。

例：　/*　1 フレーム 50ms で連続取り込みを行う場合（ランダムトリガモード）　*/

```
void frame_chaine_fvc06( void )
{
        int channel = Lib_get_video_channel();

        // 連続取り込みの 1 フレーム辺りの時間
        const int cycle_time = 50 * 1000; // 50ms

        const int cnts = 16;
        int mem_no[cnts];
        int i;

        // メモリの確保
        for( i = 0; i < cnts; i++ )
          mem_no[i] = Lib_alloc_gray_memory();

        // 連続取り込みの 1 フレーム辺りの取り込み時間を設定する。
        Lib_capt_set_reg( channel, FVC06_TRIGGER_CYCLETIME, cycle_time );

        // 連続取り込み開始
        Lib_frame_chain_freeze( mem_no[0], cnts );
}
```

Lib_camera_chain_freeze

機　能

連続フレームビデオフリーズ(カメラ切り替えあり)

形　式

```
#include "f_video.h"
int Lib_camera_chain_freeze( int start_memory_no, int frame_su );
```

901	902	903	904	FVL/LNX	高分解能
△	△	○	○		

解　説

ビデオ入力チャネルを切り替えながら画像を開始メモリ以降、取得フレーム数分取込ます。

① ***start_memory_no*** は開始メモリ番号です。

② ***frame_su*** は取得フレーム数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオ入力チャネル0, 1から1フレームずつ濃淡画像を取込、グレイメモリ0, 1へ格納します。

```
#include "f_video.h"

#define SET_CHANNEL        0
#define SET_BINARY         128
#define SET_OFFSET         0
#define START_MEMORY_NO    0
#define FRAME_SU           2

void input_video()
{
    Lib_channel( WAITING, SET_CHANNEL, SET_OFFSET, SET_BINARY );

    Lib_input_video_control( GRAY_PLANE );

    Lib_set_field_squence( SET_ANY_FIELD_CONTINUOUS );

    Lib_camera_chain_freeze( START_MEMORY_NO, FRAME_SU );

    Lib_set_field_squence( SET_ANY_FIELD );
}
```

留意事項

共通

○開始メモリ以降、取得フレーム数分の画像メモリは事前に確保しておいてください。

○外部同期でカメラを動作させてください（903を除く）。

○映像を1フレーム取得する度に、カレントのビデオ入力チャネルがインクリメントされます。
例　0-1-2-3-0-1･･･　（901AT，903，FVC01の1枚搭載では 0-1-0-1･･･ ）

○ビデオ入力数が4チャネル未満の場合について
前頁の例の様に入力チャネル0，1から連続取込みをした場合、本ライブラリ実行完了時のカレントビデオ入力チャネルは2になっていますので、チャネル2にカメラが接続されていないと一瞬画像が乱れます。もしこの画像の乱れが問題になる場合は、接続カメラの同期をとっている外部同期信号をSYNC INに入力してください。

○ビデオ入力モードは通常入力モードでのみ動作します。

○取得フレーム数が17以上の場合は、Lib_xcamera_chain_freezeを指定してください。

901

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：　(Lib_get_fx_size() + 16) * Lib_get_fy_size() * 取得フレーム数
2値画像　：　(Lib_get_fx_size() / 8 + 16) * Lib_get_fy_size() * 取得フレーム数

○事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

○Lib_input_video_controlにより、取り込む入力映像を指定できます。

○本ライブラリは、CSC901，CSC901a，CSC901SBではサポートしておりません。

902

2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：　16 + 16 * Lib_get_fy_size() * 取得フレーム数

○事前に Lib_set_field_squence ()にてフィールド・シーケンスを任意のフィールド入力に設定しておいてください。
RICEでは、Lib_set_field_squence(SET_ANY_FIELD)で御使用下さい。

○本ライブラリは、VCAP，CATS，FVC02，FVC05ではサポートしておりません。

903

○2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：　(Lib_get_fx_size() + 16) * Lib_get_fy_size() * 取得フレーム数

904

○2値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：（ Lib_get_fx_size() + 16 ） * Lib_get_fy_size() * 取得フレーム数

○事前にLib_set_field_squenceにてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

Lib_xframe_chain_freeze

機　能

連続フレームビデオフリーズ(カメラ切り替えなし)Ⅱ

形　式

```
#include "f_video.h"
int Lib_xframe_chain_freeze( char *gray_address,
                              char *bin_address, int frame_su );
```

901	902	903	904	FVL/LNX	高分解能
△	△	○	○	△	△

解　説

カレントビデオ入力チャネルの連続画像を開始アドレス以降、取得フレーム数分取込ます。Lib_frame_chain_freeze との違いは、Lib_frame_chain_freeze が画像メモリへ格納するため最大16フレームまでしか取り込めませんが、本ライブラリではユーザー指定のバッファに格納できるためメモリの許すかぎり取込が可能です。

① ***gray_address** は濃淡画像格納用バッファ開始アドレスです。

② ***bin_address** は2値画像格納用バッファ開始アドレスです。

③ ***frame_su*** は取得フレーム数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例　　10フレーム分の連続画像をユーザー指定のバッファへ取込ます。

```
#include "f_video.h"

#define FRAME_SU        10
#define MEM_NO          0

void input_video()
{
    int data_size, i;
    char *gray_address;
    char *bin_address;

    Lib_input_video_control( GRAY_PLANE );
    Lib_set_field_squence( SET_ANY_FIELD_CONTINUOUS );

    data_size = Lib_get_fx_size() * Lib_get_fy_size() * FRAME_SU;
    gray_address = (void *)Lib_mlalloc( data_size );
    bin_address  = (void *)NULL;

    Lib_xframe_chain_freeze( gray_address, bin_address, FRAME_SU );

    for ( i=0; i<FRAME_SU; i++ )
    {
        Lib_buff_to_memory( gray_address, i, GRAY_PLANE, MEM_NO );
        Lib_xvideo_transmit( MEM_NO, GRAY_PLANE );
    }

    Lib_set_field_squence( SET_ANY_FIELD );
}
```

留意事項

共通

〇ビデオ入力モードは通常入力モードでのみ動作します。

〇FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

901

〇連続画像を格納するバッファは事前に確保しておいてください。(濃淡と2値で取得したいものの
バッファを確保する)バッファの確保サイズの計算式は以下のとおりです。
濃淡画像:data_size = Lib_get_fx_size() * Lib_get_fy_size() * 取得フレーム数
2値画像 :data_size = (Lib_get_fx_size() * Lib_get_fy_size() >> 3) * 取得フレーム数

〇本ライブラリ呼び出し時さらに、Lib_get_fy_size() * 16 * 取得フレーム数の空きメモリが必
要です。

〇事前にLib_set_field_squenceにてフィールド・シーケンスを任意のフィールドから連続フレー
ム入力に設定しておいてください。

〇Lib_input_video_control により、取り込む入力映像を指定できます。
指定されなかった映像のバッファ開始アドレスは本ライブラリでは参照しないため、あえて設
定する必要はありません。

〇本ライブラリは、CSC901, CSC901a, CSC901SBではサポートしておりません。

902

〇2値画像の連続取込はFVC02またはFVC05のみの対応となります。

〇本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像: 16 + 16 * Lib_get_fy_size() * 取得フレーム数

〇事前にLib_set_field_squenceにてフィールド・シーケンスを任意のフィールド入力に設定して
おいてください。
RICEでは、Lib_set_field_squence(SET_ANY_FIELD)で御使用下さい。

〇本ライブラリは、VCAP, CATS ではサポートしておりません。

903

〇2値画像の連続取込には対応していません(bin_addressは無視されます)。

〇取得フレーム数は、最大200です。

〇本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像:(Lib_get_fx_size() + 16) * Lib_get_fy_size() * 取得フレーム数

904

○2値画像の連続取込には対応していません。

○取得フレーム数は最大200です。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：（ Lib_get_fx_size() + 16 ） * Lib_get_fy_size() * 取得フレーム数

○事前にLib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

FVL/LNX

○2値画像の連続取込はFVC02のみの対応となります。

○Lib_mlalloc 等で確保したメモリを画像取込用として使う場合は、確保後に mlock()、解放 (Lib_lfree等) 前にmunlock()をコールする必要がございます。

<例>

```
#include <sys¥mman.h>
 ・
 ・
if( (char *)NULL != ( gray_address = (void *)Lib_mlalloc( data_size ) ) )
{
    mlock(gray_address, data_size);

     /*連続フレーム入力*/


     munlock(gray_address, data_size);
}
 ・
 ・
```

○FVC06, FHC3312, FHC3313 をランダムトリガモードで使用する場合には、1 フレーム辺りの取り込み時間が Lib_capt_set_reg 関数の FVC06_TRIGGER_CYCLETEIME(FVC06 の場合)パラメータで指定した値になります。

使用方法につきましては、Lib_frame_chain_freeze の項を参照下さい。

Lib_xcamera_chain_freeze

機　能

連続フレームビデオフリーズ(カメラ切り替えあり)Ⅱ

形　式

```
#include "f_video.h"
int Lib_xcamera_chain_freeze( char *gray_address,
                              char *bin_address, int frame_su );
```

901	902	903	904	FVL/LNX	高分解能
△	△	○	○		

解　説

ビデオ入力チャネルを切り替えながら画像を開始アドレス以降、取得フレーム数分取込ます。
Lib_camera_chain_freeze との違いは、Lib_camera_chain_freeze が画像メモリへ格納するのため最大16フレームまでしか取り込めませんが、本ライブラリではユーザー指定のバッファに格納できるためメモリの許すかぎり取込が可能です。

① ***gray_address*** は濃淡映像格納用バッファ開始アドレスです。

② ***bin_address*** は2値映像格納用バッファ開始アドレスです。

③ ***frame_su*** は取得フレーム数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ビデオ入力チャネル0, 1から1フレームずつ濃淡画像を取込、ユーザー指定のバッファへ格納します。

```
#include "f_video.h"

#define FRAME_SU            2
#define MEM_NO              0

void input_video()
{
    int data_size, i;
    char *gray_address;
    char *bin_address;

    Lib_input_video_control( GRAY_PLANE );
    Lib_set_field_squence( SET_ANY_FIELD_CONTINUOUS );

    data_size = Lib_get_fx_size() * Lib_get_fy_size() * FRAME_SU;
    gray_address = (void *)Lib_mlalloc( data_size );
    bin_address  = (void *)NULL;

    Lib_xcamera_chain_freeze( gray_address, bin_address, FRAME_SU );

    for ( i=0; i<FRAME_SU; i++ )
    {
        Lib_buff_to_memory( gray_address, i, GRAY_PLANE, MEM_NO );
        Lib_xvideo_transmit( MEM_NO, GRAY_PLANE );
    }

    Lib_set_field_squence( SET_ANY_FIELD );
}
```

留意事項

共通

○ビデオ入力モードは、通常入力モードでのみ動作します。

○全てのカメラを同期させてください(903を除く)。

○映像を1フレーム取得する度に、カレントのビデオ入力チャネルがインクリメントされます。
例　　0-1-2-3-0-1･･･ (901AT、903では 0-1-0-1･･･)

○ビデオ入力数が4チャネル未満の場合について
前頁の例の様に入力チャネル0, 1から連続取込みをした場合、本ライブラリ実行完了時のカレントビデオ入力チャネルは2になっていますので、チャネル2にカメラが接続されていないと一瞬画像が乱れます。もしこの画像の乱れが問題になる場合は、接続カメラの同期をとっている外部同期信号をSYNC INに入力してください。

901

○連続画像を格納するバッファは事前に確保しておいてください。(濃淡と2値で取得したいものののバッファを確保する)バッファの確保サイズの計算式は以下のとおりです。
濃淡画像:data_size = 　Lib_get_fx_size() * Lib_get_fy_size() 　　　　* 取得フレーム数
2値画像 :data_size = (Lib_get_fx_size() * Lib_get_fy_size() >> 3) * 取得フレーム数

○本ライブラリ呼び出し時さらに、Lib_get_fx_size() * 16 * 取得フレーム数の空きメモリが必要です。

○事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

○Lib_input_video_control により、取り込む入力映像を指定できます。
指定されなかった映像のバッファ開始アドレスは本ライブラリでは参照しないため、あえて設定する必要はありません。

○本ライブラリは、CSC901, CSC901a, CSC901SB ではサポートしておりません。

902

○2 値画像の連続取込には対応していません。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像 : 16 + 16 * Lib_get_fy_size() * 取得フレーム数

○事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールド入力に設定しておいてください。
RICE では、Lib_set_field_sequence(SET_ANY_FIELD)で御使用下さい。

○本ライブラリは、VCAP, CATS, FVC02, FVC05 ではサポートしておりません。

903

○2 値画像の連続取込には対応していません(bin_address は無視されます)。
(入力ビデオに濃淡画像が選択されていないと ERROR_RETURN します)。

○本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像 : (Lib_get_fx_size() + 16) * Lib_get_fy_size() * 取得フレーム数

〇2値画像の連続取込には対応していません。

〇取得フレーム数は最大200です。

〇本ライブラリ呼び出し時、下記の空きメモリが必要です。
濃淡画像：(Lib_get_fx_size() + 16) * Lib_get_fy_size() * 取得フレーム数

〇事前に Lib_set_field_squence にてフィールド・シーケンスを任意のフィールドから連続フレーム入力に設定しておいてください。

Lib_get_last_field_id

機　能

取得フィールド番号の参照

形　式

```
#include "f_video.h"
int Lib_get_last_field_id( void );
```

901	902	903	904	FVL/LNX	高分解能
△	△		△		○

解　説

任意のフィールドからの1フィールド取込、またはランダムトリガ入力で取り込んだ時のフィールド番号を取得します。

戻り値

直近に取得したフィールド。

値	定　　数	意　　味
0	ID_1ST_FIELD	第1フィールド
1	ID_2ND_FIELD	第2フィールド

例

任意のフィールドから1フィールド取込時の取得フィールド番号を参照する。

```
#include "f_video.h"
#include "f_stdLib.h"
#include "f_stdio.h"

void video_input()
{
    int field_no;

    /* 任意フィールドから1フィールド取得 */
    Lib_set_field_squence( SET_ANY_SINGLE_FIELD );

    Lib_memory_clear( GRAY_PLANE );
    Lib_freeze( NOT_TRANSMIT );

    field_no = Lib_get_last_field_id( );

    Lib_sprintf( ss, "field_no = %d", field_no );
    Lib_chrdisp( 1, 28, ss );
    Lib_xvideo_transmit( 0, GRAY_PLANE );
    Lib_display_keyinput( 430, 0, "  ANY  " );
}
```

留意事項

901，904

〇Lib_set_field_squence で、第1フィールドからもしくは第2フィールドからの指定で1フィールド取り込んだ場合、本ライブラリは使用できません。

〇本ライブラリは、FV904p，FV904pc，CSC901，CSC901a，CSC901SBではサポートしておりません。

902

〇本ライブラリは、VCAP，FVC01，FVC02，FVC05ではサポートしておりません。

Lib_2ch_indication

機　能

2CH同時表示

形　式

```
#include "f_video.h"
int Lib_2ch_indication( int mem_no, int L_channel, int R_channel, int mode );
```

901	902	903	904	FVL/LNX	高分解能
○	○	○	○	○	△

解　説

任意の2カメラから画像を取込1モニターの上半分に(1カメラ1/4画像で)同時表示します。

① ***mem_no*** は表示用画像メモリ番号
[注]グレイ画像の時は、グレイのメモリ番号を2値画像の時は2値のメモリ番号を設定してください。

② ***L_channel*** は同時表示時の左側に表示するチャネルNO
範囲は0～3(901AT、903では0～1)の数値で指定してください

③ ***R_channel*** は同時表示時の右側に表示するチャネルNO
範囲は0～3(901AT、903では0～1)の数値で指定してください

④ ***mode*** はグレイ＆2値どちらを2CH表示するか選択してください

値	定　　数	意　　味
0x01	BIN_PLANE	2値用フレームバッファを指定します。
0x20	GRAY_PLANE	濃淡用フレームバッファを指定します。

戻り値

処理結果

値	定　　数	意　　味
0	ありません。	正常終了しました。
-1	ありません。	異常終了しました。
-2	ありません。	パラメータ入力エラー

例

チャネル0と1からグレイ画像を取込、2CH同時表示をします

```
#include "f_video.h"
#include "f_stdLib.h"
#include "f_gui.h"

void set_camera()
{
    int x, y;
    int process;
    int mem_no;

    mem_no = Lib_alloc_gray_memory();

    for( process = ON; process == ON; )
    {
        Lib_2ch_indication( mem_no, 0, 1, GRAY_PLANE );

        if( CURSOR_CANCEL == Lib_see_current_position( &x, &y ) )
            process = OFF;
    }

    Lib_free_gray_memory( mem_no );
}
```

留意事項

○モニタTVの下半分の画像データは「黒」で表示されます。

○表示用画像メモリ番号にカレントメモリ番号を指定する事はできません。

○本ライブラリ内では、チャネル切り換え、画像取込、モニタ画面の表示までを同時に行っています。

○外部同期でカメラを動作させてください。

○FVL/LNXをご使用の場合のみ、高分解能カメラをサポートします。

Lib_xfreeze

機　能

拡張ビデオ フリーズ

形　式

```
#include "f_video.h"
int Lib_xfreeze( void );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	○

解　説

最新の入力ビデオをカレントメモリ上に凍結します。
フリーラン中に起動した場合は、モニタ表示は静止画状態となります。
フリーズ中に起動した場合は、モニタ表示は変わりません。
本ライブラリは Lib_freeze と違い、起動後、画像メモリに入力ビデオの取込が完了する前に終了します。
取込の完了は Lib_get_video_status で取得してください。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

①画像の入力をします。

```
#include "f_video.h"


void xfreeze()
{
    Lib_input_video_control( GRAY_PLANE );

    Lib_xfreeze();
    while( FINISHED_TRANSMIT != Lib_get_video_status
                                    ( Lib_get_video_channel() ) )
    {
        /* 取込が完了するまで別処理をします */
    }
}
```

留意事項

○本ライブラリを起動後、取込が完了する前に、
Lib_freeze ,Lib_freerun ,Lib_xfreeze ,
Lib_set_video_input_mode ,Lib_channel等の
ビデオ入力制御ライブラリは使用できません。

○本ライブラリは濃淡画像のみ有効で、2値画像には使用できません。
ビデオ入力制御は濃淡画像が選択されている必要があります。

○ビデオ入力用現在チャネル(システムパラメータ)
システムパラメータ「ビデオ入力用現在チャネル」により、ビデオデータを入力するチャネルが指定されます。
Lib_channel,Lib_video_channel,Lib_stage_input,Lib_set_stage_inputにより、指定できます。

○ビデオ入力オフセット値(システムパラメータ)
システムパラメータ「ビデオ入力オフセット値」にて入力ビデオのオフセット方式が指定されます。
Lib_channel, Lib_video_offsetにより、指定できます。

○ビデオ入力制御(システムパラメータ)
システムパラメータ「ビデオ入力制御」にて入力ビデオの映像が指定されます。
Lib_input_video_controlにより、入力映像を指定できます。

○メモリ
☆濃淡画像メモリ8ビット全体に取込まれます。

○モニタTV表示内容
☆フリーラン中の起動
VCAP,CATS : 最大で1フレーム前後します。
RICE,FVC01 : 画像メモリ内容の1フレーム前のものとなります。
☆フリーズ中の起動
モニタTV表示画面はかわりません。

○本ライブラリは、CATSではサポートしておりません。

Lib_get_video_status

機　能

ビデオ入力取込み完了ステータスの取得

形　式

```
#include "f_video.h"
int Lib_get_video_status ( int channel );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	○

解　説

指定したチャネル番号のビデオ入力取込完了ステータスを参照します。

① ***channel*** はチャネル番号です。(0～3)

戻り値

取込完了ステータスを返します。

値	定　　数	意　　　味
0	FINISHED_TRANSMIT	取込が完了しました。
1	TRANSMITING	取込中です。

例

画像の入力後、取込み完了まで待ちます。

```
#include "f_video.h"


void get_video_status()
{
    Lib_input_video_control( GRAY_PLANE );

    Lib_xfreeze( TRANSMIT );
    while( FINISHED_TRANSMIT != Lib_get_video_status
                                        ( Lib_get_video_channel ) )
    {
        /*   何らかの処理  */
    }
}
```

留意事項

○本ライブラリは Lib_xfreeze 後、TRANSMITING状態になり、取込が完了すると FINISHED_TRANSMIT 状態になります。
Lib_xfreezeでエラーが発生した場合は、FINISHED_TRANSMIT状態になります。
ほかのライブラリ(Lib_freeze 等)との組み合わせでは使用できませんので、ご注意下さい。

○本ライブラリは、CATSではサポートしておりません。

○RICE の場合引数のチャネル番号は意味を持ちません。カレント･チャネルのステータスが返ります。

Lib_set_sim_channel

機　能

ビデオ同時取込に使用するチャネル設定

形　式

```
#include "f_video.h"
int Lib_set_sim_channel( int channel_flag );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	

解　説

ビデオ同時取込に使用するチャネルを指定します。
ビデオ入力モードが、
SIM_VIDEO_MODE
RANDOM_XC_SIM_MODE
RANDOM_XC_EX_SIM_MODE
時に使用します。

① ***channel_flag*** はチャネル番号フラグです。
ビットパタンで複数指定可能です。

値	定　数	意　味
0x01	SIM_CH0	チャネル0
0x02	SIM_CH1	チャネル1
0x04	SIM_CH2	チャネル2
0x08	SIM_CH3	チャネル3
0x10	SIM_CH4	チャネル４（ FV-GP440 にて対応 ）
0x20	SIM_CH5	チャネル５（ FV-GP440 にて対応 ）
0x40	SIM_CH6	チャネル６（ FV-GP440 にて対応 ）
0x80	SIM_CH7	チャネル７（ FV-GP440 にて対応 ）

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

①チャネル0, 1からグレイ画像の同時取込を行います。

```
#include "f_video.h"

#define  CHANNEL0    0
#define  CHANNEL1    1

void sim_freeze()
{
    int  iMemNo0, iMemNo1;

    Lib_input_video_control( GRAY_PLANE );
    Lib_set_video_input_mode( SIM_VIDEO_MODE, 0 );
    iMemNo0 = Lib_alloc_gray_memory();
    iMemNo1 = Lib_alloc_gray_memory();

    Lib_set_sim_channel( SIM_CH0 | SIM_CH1 );
    Lib_set_sim_gray_memory( CHANNEL0, iMemNo0 );
    Lib_set_sim_gray_memory( CHANNEL1, iMemNo1 );

    Lib_freeze( TRANSMIT );

    /* 処理 */

}
```

②チャネル0, 1から2値画像の同時取込を行います。

```
#include "f_video.h"

#define  CHANNEL0    0
#define  CHANNEL1    1

void sim_freeze()
{
    int  iMemNo0, iMemNo1;

    Lib_input_video_control( BIN_PLANE );
    Lib_set_video_input_mode( SIM_VIDEO_MODE, 0 );

    iMemNo0 = Lib_alloc_bin_memory();
    iMemNo1 = Lib_alloc_bin_memory();
    Lib_set_sim_channel( SIM_CH0 | SIM_CH1 );
    Lib_set_sim_bin_memory( CHANNEL0, iMemNo0 );
    Lib_set_sim_bin_memory( CHANNEL1, iMemNo1 );

    Lib_freeze( TRANSMIT );

    /* 処理 */

}
```

留意事項

○本ライブラリはCATSではサポートしておりません。

○902シリーズでRICEをご使用の場合、
ビデオ同時取込をサポートしていますが、本関数の設定に関係なく、
Lib_set_video_input_mode でチャネル番号に0または1を指定した場合は0と1チャネルの、2又は3を指定した場合は2と3チャネルの2チャネル同時取込を行います。

○LNXシリーズでRICEをご使用の場合
本関数で同時取込に使用するチャネルを設定します。
Lib_set_video_input_mode で設定した値は無効となります。

○FVC02, FVC05及びFV-GP440でビデオ同時取込をキャンセルする場合は、
チャネル番号フラグに0を指定してください。

Lib_set_sim_gray_memory

機　能

同時取込に使用するグレイメモリ番号の指定

形　式

```
#include "f_video.h"
int Lib_set_sim_gray_memory( int channel, int memory );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	

解　説

同時取込に使用するグレイメモリ番号を指定します。

① ***channel*** はチャネル番号です。(0～3)

② ***memory*** はグレイメモリ番号です。
　-1を指定した場合カレントグレイメモリになります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

Lib_set_sim_channelの例を参照してください。

留意事項

○本ライブラリは、CATSではサポートしておりません。

○ビデオ入力モードを Lib_set_video_input_mode で設定する場合は、
　　SIM_VIDEO_MODE
　　RANDOM_XC_SIM_MODE
　　RANDOM_XC_EX_SIM_MODE
　時に有効です。

Lib_set_sim_bin_memory

機　能

同時取込に使用する2値メモリ番号の指定

形　式

```
#include "f_video.h"
int Lib_set_sim_bin_memory( int channel, int memory );
```

901	902	903	904	FVL/LNX	高分解能
	△			○	

解　説

同時取込で使用する2値メモリ番号を指定します。

① ***channel*** はチャネル番号です。(0～3)

② ***memory*** は2値メモリ番号です。
　-1を指定した場合カレント2値メモリになります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

Lib_set_sim_channelの例を参照してください。

留意事項

○本ライブラリはCATSではサポートしておりません。

○ビデオ入力モードを Lib_set_video_input_mode で設定する場合は、
　　SIM_VIDEO_MODE
　　RANDOM_XC_SIM_MODE
　　RANDOM_XC_EX_SIM_MODE
　時に有効です。

Lib_set_exposure

機　能

露光時間設定

形　式

```
#include "f_video.h"
int Lib_set_exposure(int time);
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△	△	

解　説

ランダムトリガモード時の露光時間を設定します。

① *time* は露光時間です（μ秒）。設定範囲は留意事項を参照して下さい。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ランダムトリガモードに移行し、露光時間・2値化レベル設定後に画像を取込ます。

```
#include "f_video.h"
#include "f_time.h"

#define CH_0            0       /* チャネル 0 */
#define DELAY_RANDOM    200     /* ランダムトリガモード移行後の
                                                      待ち時間 ms */
#define EXPOSURE        33000   /* 露光時間 μs */
#define BIN_LEVEL       128     /* ランダムトリガ2値化レベル */


void random_freeze()
{

    Lib_input_video_control(BIN_PLANE);
    Lib_display_control(BIN_PLANE|LINE_PLANE|CHAR_PLANE);
    Lib_video_channel(CH_0);
    Lib_set_video_input_mode(RANDOM_XC_MODE,CH_0);
    Lib_set_exposure(EXPOSURE);
    Lib_random_bin_level(BIN_LEVEL);

    /* ランダムトリガモードに移行後の最初の画像取込までには
                                                      ウエイトを入れる */
    Lib_time_delay(DELAY_RANDOM);

    Lib_freeze(TRANSMIT);
}
```

留意事項

902

○本ライブラリは、FVC01，FVC02，FVC05で使用可能です。

○FVC02またはFVC05でTI-480Aをご使用の場合、本関数で露光時間の設定はできません。
カメラ本体側で設定します。

○露光時間は、カレントビデオ入力チャネル番号に対応するボードの2CHに設定されます。

○設定された露光時間は、再設定されるまで保持されます。

○FVC01の設定範囲
設定範囲は10μs～250000μs(250ms)です。
10～60μsは2μs刻みで、それ以上は4μs刻みで露光時間が設定されます。

○FVC02の設定範囲
10μs～655350μsです。

○FVC05の設定範囲
1μs～3276750μsです。

903，904

○本ライブラリはCSC903，FV904p，FV904pcのみで使用可能です。

○露光時間は、各チャネル共通です。

○設定された露光時間は、再設定されるまで保持されます。

○内部でLib_set_random_exposureがコールされるのでシステムパラメータも更新されます。
(Lib_set_random_exposure に関しては、「90X基本ライブラリ説明書 Vol.3」をご参照下さい。)

○設定範囲
10μs～250000μsです。
10～60μs刻みで、それ以上は4μs刻みで露光時間が設定されます。

FVL/LNX

○本ライブラリは FVC02, FVC04, FVC05, FV1100-LNX, FVC06, FV-GP440, FHC3312, FHC3313 で使用可能です。

○FVC02, FVC05でTI-480AおよびTI-1200Aをご使用の場合、本関数で露光時間の設定はできません。カメラ本体側で設定します。

○露光時間は、カレントビデオ入力チャネル番号に対応するボードの2CHに設定されます。
※FVC06の場合にはカレントビデオ入力チャネルにのみ設定されます。
※FV-GP440 の場合にはカレントビデオ入力チャネルにのみ設定されます。
※FHC3312, FHC3313の場合にはカレントビデオ入力チャネルにのみ設定されます。

○設定された露光時間は再設定されるまで保持されます。

○FVC02の設定範囲
　10μs～250000μsです。

○FVC05 の設定範囲
　1μs～3276750μsです。

○FVC06 の設定範囲
　1μs～64424509μs です。

○FV-GP440 の設定範囲
　1μs～3276750μs です。

○FHC3312, FHC3313 の設定範囲
　1μs～2147483647μs です。

Lib_get_exposure

機　能

露光時間参照

形　式

```
#include "f_video.h"
int Lib_get_exposure();
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△	○	

解　説

ランダムトリガモード時の露光時間を参照します。

戻り値

露光時間です（μ秒）。

留意事項

902

○本ライブラリは、FVC01，FVC02，FVC05で使用可能です。

○露光時間は、カレントビデオ入力チャネル番号に対応するボードの設定です。

903，904

○本ライブラリはCSC903，FV904p，FV904pcのみで使用可能です。

○露光時間は、各チャネル共通です。

○設定された露光時間は再設定されるまで保持されます。

FVL/LNX

○本ライブラリは、FVC02，FVC05，FVC06，FV-GP440，FHC3312，FHC3313で使用可能です。

○露光時間は、カレントビデオ入力チャネル番号に対応するボードの設定です。

Lib_random_bin_level

機　能

ランダムトリガ2値化レベル設定

形　式

```
#include "f_video.h"
int Lib_random_bin_level(int bin_level);
```

901	902	903	904	FVL/LNX	高分解能
		○	△		○

解　説

ランダムトリガモード時の2値化レベルを設定します。

① ***bin_level*** は2値化レベル(0～255)です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

Lib_set_exposure関数の例を参照してください。

留意事項

○本ライブラリはCSC903, FV904p, FV904pcのみで使用可能です。

○2 値化レベルは各チャネル固有です。カレントチャネルの 2 値化レベルが設定されます。

○設定された 2 値化レベルは、再設定されるまで保持されます。

○内部で Lib_set_random_bin_level 関数がコールされるので、システムパラメータも更新されます。(Lib_set_random_bin_level に関しては、「90X 基本ライブラリ説明書 Vol.3」をご参照下さい。)

Lib_set_both_mode

機　能

同時露光モード設定

形　式

```
#include "f_video.h"
int Lib_set_both_mode(unsigned int sw, int ch, int no0, int no1);
```

901	902	903	904	FVL/LNX	高分解能
		○			

解　説

ランダムトリガモードにて2チャネル同時露光モードに設定します。

① ***sw*** は取込画像スイッチです(OR設定)。
(該当する取込画像のビットをオンにしたビットパタンで指定します)

値	定　　数	意　　味
0x01	RANDOM_TRG_SW_GRAY	濃淡画像取込
0x02	RANDOM_TRG_SW_BIN	2値画像取込

② ***ch*** は取込優先チャネルです。設定範囲は0～1です。
優先チャネルから先に画像をメモリに取込ます。
フレームバッファへ画像を転送する場合は優先チャネルの画像が転送されます。

③ ***no0*** はチャネル0画像取込先メモリ番号です。
sw の RANDOM_TRG_SW_GRAY ビットがONならば、ここで指定された濃淡メモリへ画像が転送されます。
sw の RANDOM_TRG_SW_BIN ビットがONならば、ここで指定された2値メモリへ画像が転送されます。

④ ***no1*** はチャネル1　画像取込先メモリ番号です。
sw の RANDOM_TRG_SW_GRAY ビットがONならば、ここで指定された濃淡メモリへ画像が転送されます。
sw の RANDOM_TRG_SW_BIN ビットがONならば、ここで指定された2値メモリへ画像が転送されます。

ビデオ入力モードが　RANDOM_XC_MODE_BOTH_CH
XC55 ランダムトリガ・2 チャネル同時モードに設定されます。
設定後に画像取込(Lib_freeze 等)を実行すると、チャネル 0 とチャネル 1 の両方で同時に露光が行われます。露光結果は ch(取込優先チャネル)で指定された優先順位に従い、両チャネルとも画像メモリへ転送されます。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

2チャネル同時露光を行ってチャネル0、チャネル1の順に画像を表示します。

```
#include "f_video.h"
#include "f_time.h"
#include "f_gui.h"

#define CH_0            0       /* チャネル 0 */

void random_both_freeze()
{
    int no0,no1;

    Lib_init_cursor();
    Lib_input_video_control(GRAY_PLANE);
    Lib_display_control(GRAY_PLANE|LINE_PLANE|CHAR_PLANE);
    Lib_video_channel(CH_0);
    no0 = Lib_get_gray_memory();
    no1 = Lib_alloc_gray_memory();
    Lib_set_both_mode(RANDOM_TRG_SW_GRAY,CH_0, no0, no1);

    /* 画像取込 */
    Lib_freeze(TRANSMIT);
    Lib_display_message(10,10,"チャネル０画像","確認をクリックして下さい");

    Lib_xvideo_transmit(no1,GRAY_PLANE);
    Lib_display_message(10,10,"チャネル１画像","確認をクリックして下さい");

    Lib_set_video_input_mode(NORMAL_VIDEO_MODE,CH_0);

    Lib_freerun();
    Lib_free_gray_memory(no1);
}
```

留意事項

ありません。

Lib_ena_ext_trg

機　能

外部トリガ信号待ち

形　式

```
#include "f_video.h"
int Lib_ena_ext_trg(int flag);
```

901	902	903	904	FVL/LNX	高分解能
		○	△		

解　説

ランダムトリガモードにて外部トリガ信号を待ちます。
FREEZE_COMPLETEが返されるまで繰り返しコールしてください。

① ***flag*** はフレームバッファへの転送フラグです。
TRANSMIT　　　転送する
NOT_TRANEMIT　　転送しない

戻り値

処理結果

値	定　数	意　味
0	WAITING_TRG	外部トリガ待ち状態です。
1	FREEZE_COMPLETE	画像取込を完了しました。
-1	ERROR_RETURN	異常終了しました。

例

外部トリガにて画像を取込ます。

```
#include "f_video.h"
#include "f_time.h"

#define CH_0            0       /* チャネル 0 */

void random_ext_trg()
{
    int return_code;

    Lib_input_video_control(GRAY_PLANE);
    Lib_display_control(GRAY_PLANE|LINE_PLANE|CHAR_PLANE);
    Lib_video_channel(CH_0);
    Lib_set_video_input_mode(RANDOM_XC_MODE,CH_0);

    do
    {
        return_code = Lib_ena_ext_trg(TRANSMIT);
    } while(return_code == WAITING_TRG);
}
```

留意事項

○「画像取込中」という戻り値は存在しません。
　画像取込が始まった場合は、完了までこの関数から戻らないからです。

○本ライブラリは CSC903, FV904p, FV904pc のみで使用可能です。

Lib_dis_ext_trg

機　能

外部トリガ信号待ち中止

形　式

```
#include "f_video.h"
int Lib_dis_ext_trg(void);
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△	△	○

解　説

ランダムトリガモードにて外部トリガ信号待ちを中止します。
903では信号待ちを中止した時点で既にトリガを受けて画像取込済みの場合は、FREEZE_COMPLETE を返します。
FVC01, FVC02, FVC05, FVC06, FV-GP440, FHC3312, FHC3313 では、画像取込済みの場合でも DISABLED_TRG を返します。

戻り値

処理結果

値	定　　数	意　　味
0	DISABLED_TRG	外部トリガ信号待ちを中止しました。
1	FREEZE_COMPLETE	画像取込を完了しました。
-1	ERROR_RETURN	異常終了しました。

例

CANCEL キーが押された場合は外部トリガ信号待ちを中止します(903での例)。

```
#include "f_video.h"
#include "f_time.h"
#include "f_gui.h"

#define CH_0            0    /* チャネル 0 */
#define DISP_X          1    /* 状態表示位置X */
#define DISP_Y          25   /* 状態表示位置Y */

void random_dis_ext_trg()
{
    int return_code;
    int x = 0, y = 0;

    Lib_init_cursor();
    Lib_input_video_control(GRAY_PLANE);
    Lib_display_control(GRAY_PLANE|LINE_PLANE|CHAR_PLANE);
    Lib_video_channel(CH_0);
    Lib_set_video_input_mode(RANDOM_XC_MODE,CH_0);

    while(1)
    {
        return_code = Lib_ena_ext_trg(TRANSMIT);
        if(return_code == WAITING_TRG)
        {
```

```
                    /* CANCEL キーが押された場合は外部トリガ信号待ちを中止 */
                    if(Lib_see_current_position(&x,&y) == CURSOR_CANCEL)
                    {
                        return_code = Lib_dis_ext_trg();
                        if(return_code == DISABLED_TRG)
                        {
                            Lib_chrdisp(DISP_X,DISP_Y,"画像取込中止  ");
                            break;
                        }
                    }
                }
                if(return_code == FREEZE_COMPLETE)
                {
                    Lib_chrdisp(DISP_X,DISP_Y,"画像取込完了  ");
                    break;
                }
                if(return_code == ERROR_RETURN)
                {
                    Lib_chrdisp(DISP_X,DISP_Y,"異常終了      ");
                    break;
                }
                Lib_chrdisp(DISP_X,DISP_Y,"外部トリガ待ち");
            }
            Lib_time_delay(5000);
            Lib_chrdisp(DISP_X,DISP_Y,"              ");
        }
```

留意事項

902

○FVC01, FVC02, FVC05でLib_xfreeze関数をコールした場合に利用して下さい。

○カレントビデオ入力チャネル番号のボード内の2CHとも中止します。

903, 904

○本ライブラリは CSC903, FV904p, FV904pc のみで使用可能です。

○Lib_ena_ext_trg 関数をコールした場合に利用して下さい。

FVL/LNX

○FVC02, FVC04, FVC05, FHC3321(A), FVC06, FV-GP440, FHC3312, FHC3313でLib_xfreeze関数をコールした場合に利用して下さい。

○FVC02, FVC05 では、カレントビデオ入力チャネル番号のボード内の 2CH とも中止します。

Lib_capt_grab_abort

機　能

取込の中止

形　式

```
#include "f_video.h"
int Lib_capt_grab_abort( int channel )
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

取込を強制終了します。
Lib_dis_ext_trg 関数と違い、本関数ではどの取込モードが選択されていても途中で取り込みを終了します。
本関数は、下記のキャプチャボードで使用可能です。

- FVC05
- FV-GP440
- FVC06
- FHC3312
- FHC3313

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。 - 実行エラー - channel が異常 - キャプチャボードが対応していない

例 Lib_xfreezeを実行中に取込をキャンセルします。

```
#define my_channel 0

void sample( void )
{
  // 取込開始
  Lib_xfreeze();

  // 取込完了まで待機
  while(1)
  {
    // 取込が完了したら抜ける
    if( FINISHED_TRANSMIT == Lib_get_video_status( my_channel ) )
        break;

    // もしも取込中に何かボタンが押下されたら、
    // 取込を中止する
    if( Lib_see_current_position( NULL, NULL ) )
    {
        Lib_capt_grab_abort( my_channel );
        break;
      }
  }
}
```

留意事項 ありません

Lib_set_vline

機　能

画像垂直取込ライン設定

形　式

```
#include "f_video.h"
int Lib_set_vline(int start, int size);
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△	△	

解　説

画像の垂直取込開始ラインと取込ライン数を設定します。

① ***start*** は取込開始ラインです。
　設定範囲は留意事項を参照してください。

② ***size*** は取込ライン数です。
　設定範囲は留意事項を参照して下さい。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

垂直取込ライン設定を行います。

```
#include "f_video.h"
#include "f_time.h"
#include "f_graph.h"

#define CH_0            0       /* チャネル 0 */
#define LINE_SET_WAIT   66      /* ライン設定の待ち時間 */

long vstart,vsize;

void exec_set_line(int menu_no,int exec_timing,int n,PARADIGM *val)
{
    Lib_cls(GRAY_PLANE,0);
    switch(menu_no)
    {
        case 0:
            vstart = val[0].long_type;
            Lib_set_vline((int)vstart,(int)vsize);
            break;
        case 1:
            vsize = val[0].long_type;
            Lib_set_vline((int)vstart,(int)vsize);
            break;
        default:
            break;
```

```
    }
    Lib_time_delay(LINE_SET_WAIT);  /* ライン設定の待ち時間 */
}


void menu_set_line()
{
    int no,pad_level;
    const int pdx = 5, pdy = 30;
    PVAL value[2];

    vstart = 0;
    vsize = 480;

    Lib_input_video_control(GRAY_PLANE);
    Lib_display_control(GRAY_PLANE | LINE_PLANE | CHAR_PLANE);
    Lib_video_channel(CH_0);
    Lib_freerun();

    pad_level = Lib_view_open();
    Lib_view_set_title(pad_level,"取込ライン");

    Lib_view_set_uniq_numeral(pad_level,pdx,pdy * 0 + 5,
                                          "垂直開始",vstart,-21L,225L,0);
    Lib_view_set_uniq_numeral(pad_level,pdx,pdy * 1 + 5,
                                          "垂直長",vsize,1L,480L,1);

    Lib_set_xparadigm(pad_level,0,ST_AND_PR_TIMING,exec_set_line);
    Lib_set_xparadigm(pad_level,1,ST_AND_PR_TIMING,exec_set_line);

    Lib_view_set_size(pad_level,50,80,5,5 );

    Lib_draw_menu(pad_level );
    Lib_process_menu(pad_level,value);
    Lib_erase_menu(pad_level);
    Lib_view_close(pad_level);

    Lib_set_vline(0,480);
}
```

FVL/LNXをお使いの場合

```
#include "f_video.h"

#define SET_VILINE      480

void set_vline_freeze( void )
{
        int new_mem;

        Lib_set_vline( 0, SET_VLINE );

        /* 変更後のサイズでメモリを確保する */
        new_mem = Lib_alloc_gray_memory();
        Lib_change_gray_memory( new_mem );
```

```
		/* 取込 */
		Lib_freeze();
	}
```

留意事項

902

○FVC01

・通常入力モードでは、取込ライン数により画像取込時間は短縮されません。

・start + size は 493 以下である必要があります。

・通常入力モードでは、画像取込時間が短縮されません。

・0～479 を超えたラインでの画像存在有無については保証できません。実際にカメラを接続して確認してください。

・取込開始ラインの設定範囲は、-20～225 です(初期値 0)。

・取込ライン数の設定範囲は、1～480 です(初期値 480)。

○FVC02

・取込ライン数が 16 の倍数である必要があります。
・取込開始ライン、取込ライン数の設定範囲は以下のようになります。
取込開始ライン+取込ライン数は、取込開始ラインの設定範囲内の必要があります。

カメラ	取込開始ライン	取込ライン数
XC-55	-9 ～ 485	1 ～ 480
CS-8530-01	-9 ～ 485	1 ～ 480
CS-8530D-01	-9 ～ 485	1 ～ 480
CS-3720	0 ～ 480	1 ～ 480

903, 904

○本ライブラリはCSC903, FV904p, FV904pcのみで使用可能です。
○start + size は494以下である必要があります。
○通常入力モードでは、画像取込時間が短縮されません。
○0～479を超えたラインでの画像存在有無については保証できません。実際にカメラを接続して確認してください。
○取込開始ラインの設定範囲は、-21～225です(初期値480)。
○取込ライン数の設定範囲は、1～480です(初期値0)。
○0～479を超えたラインでの画像存在有無については保証できません。実際にカメラを接続して確認してください。
○FV904p, FV904pcで本ライブラリを使用する場合、直後に必ず空取込を1回入れて下さい。

＜設定例＞

```
Lib_set_vline(iVoffset, iVsize);
Lib_freeze(NOT_TRANSMIT); /* 空取込 */
```

○903, FV904p, FV904pcではXC-55/56で、ほぼカメラの中心部分が取り込まれる様にタイミングを設定しています。よってXC-55/56とタイミングの異なるカメラは、本ライブラリで調整が必要です。

XC-55/56とタイミングの異なるカメラ

[903対応カメラ]
CS8550i-01, CS8550Di-01 ・ ・・・・・・ +12

[904p対応カメラ]
CS8550i-01, CS8550D/CS8550Di ・・・・・ +12
KP-F30,KP-F33 ・・・・・ -6

FVL/LNX

○FVC02

・取込ライン数が16の倍数である必要があります。
・取込開始ライン、取込ライン数の設定範囲は以下のようになります。
取込開始ライン+取込ライン数は、取込開始ラインの設定範囲内の必要があります。

カメラ	取込開始ライン	取込ライン数
XC-55	-9 ～ 485	1 ～ 480
CS-8530-01	-9 ～ 485	1 ～ 480
CS-8530D-01	-9 ～ 485	1 ～ 480
CS-3720	0 ～ 480	1 ～ 480

○FVC04では、Lib_cl_setvpartを使用してください。

Lib_set_hline

機　能

画像水平取込ライン設定

形　式

```
#include "f_video.h"
int Lib_set_hline( int iStart, int iSize );
```

901	902	903	904	FVL/LNX	高分解能
	△		△	△	

解　説

画像の水平取込開始ラインと取込ライン数を設定します。

① ***iStart*** は取込開始ラインです。
　設定範囲は、カメラにより異なります。留意事項を参照してください。

② ***iSize*** は取込ライン数です。
　設定範囲は、カメラにより異なります。留意事項を参照してください。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

留意事項

902

○本関数は、FVC02で使用可能です。
○FVC02 での取込開始ライン、取込ライン数の設定範囲は以下のようになります。
取込開始ライン+取込ライン数は、取込開始ラインの設定範囲内の必要があります。

カメラ	取込開始ライン	取込ライン数
XC-55	-66 ～ 581	1 ～ 512
CS-8530-01	-66 ～ 581	1 ～ 512
CS-8530D-01	-66 ～ 581	1 ～ 512
CS-3720	-67 ～ 581	1 ～ 512

904

○本関数は、FV904p，FV904pc で使用可能です。
○取込開始ライン設定可能範囲は、-200 ～ 68 です。ただし画像の存在有無については保証できません。実際にカメラを接続して御確認下さい。
○取込ライン数は 512 のみサポートしています。512 以外を指定すると異常終了となります。
○FV904p，FV904pc で本ライブラリを使用する場合、直後に必ず空取込を 1 回入れて下さい。

＜設定例＞

```
Lib_set_hline(iVoffset, iVsize);
Lib_freeze(NOT_TRANSMIT); /* 空取込 */
```

Lib_capt_set_scanrate

機　能

スキャンレートの設定

形　式

```
#include "f_video.h"
int  Lib_capt_set_scanrate( int  channel,  int scanrate );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

スキャンレートを設定します。
使用カメラがラインセンサカメラの場合には、１ラインの取込時間になります。

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

① *channel* は設定するチャンネル番号です。

② ***scanrate*** は設定するスキャンレートです。（ 単位： μsec ）
設定範囲はカメラにより異なります。カメラの仕様書をご確認ください。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	***scanrate***がマイナスの値です。

例

スキャンレートを 180us に設定します。

```
int set_scanrate( void )
{
  int channel = Lib_get_video_channel();
  Lib_capt_set_scanrate( channel, 180 );
}
```

留意事項

ありません。

Lib_capt_get_scanrate

機　能

スキャンレートの取得

形　式

```
#include "f_video.h"
int  Lib_capt_get_scanrate( int channel, int *scanrate );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

設定されているスキャンレート値を取得します。

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

① *channel* はスキャンレート値を取得するチャンネル番号です。
② ***scanrate*** は取得したスキャンレート値です。(単位: μsec)

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	***scanrate***がNULLです。

例

留意事項

ありません。

Lib_capt_set_encoder

機　能　　エンコーダの設定

形　式

```
#include "f_video.h"
int  Lib_capt_set_encoder( int channel, int start, int mode, int dir, int signal,
                           int mlt, int cmp1, int cmp2 );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　　エンコーダを使用した取込設定をします。

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

本関数でエンコーダのパラメータを設定し、取込を行う場合には、取込モードが ENCODER_XVIDEO_MODE になっている必要があります。

① *channel* は設定するチャンネル番号です。

② ***start*** はエンコーダの開始方法です。

値	意	味
0	CPU	：画像入力の指示を受け次第、エンコーダのカウントを始めます。
1	外部トリガ	：画像入力の指示を受けると、外部トリガの入力を待ちます。外部トリガの入力を検出すると、エンコーダのカウントを始めます。
2	Z相	：画像入力の指示を受けると、Z相の入力を待ちます。Z相の入力を検出すると、エンコーダのカウントを始めます。

③ ***mode*** は取込モードです。
本パラメータは、下記キャプチャボード使用時のみ有効になります。
下記ボード以外では、設定されたパラメータは無視されます。
- FHC3312
- FHC3313

値	意　　味
0	エンコーダスキャンモード
1	エンコーダライン選択モード

④ ***dir*** はエンコーダの回転方向です。

値	意　　　味
0	CW
1	CCW

⑤ ***signal*** は検出するエンコーダの位相です。

値	意　　　味
0	AB相
1	A相

⑥ ***mlt*** はエンコーダのサンプリングレートです。

値	意　　　味
0	1倍
1	2倍
2	4倍　※***signal***にAB相を指定したときのみ使用できます。

⑦ ***cmp1*** は比較レジスタ１の値です。
　本パラメータに設定されたパルスのカウント数分だけ待機した後、取込を開始します。
－０ 以上の値を入力してください。

⑧ ***cmp2*** は比較レジスタ２の値です。
　本パラメータに設定された値とパルスカウンタが一致した場合にライントリガを出力します。1ライン分のパルス数です。
－１ 以上の値を入力してください。

戻り値

処理結果

値	定　　数	意　　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例　エンコーダの各パラメータ設定をし、取込を行います。

```
void set_encoder( int channel )
{
  int start = 0;  // CPU
  int mode = 0;   // エンコーダスキャンモード
  int dir = 0;    // CW
  int signal = 0; // AB相
  int mlt = 0;    // 1倍
  int cmp1 = 0;   // 待機しない
  int cmp2 = 20;  // エンコーダ信号が20pulse入力されたら1ラインの取込を行う

  Lib_capt_set_encoder( channel, start, mode, dir, signal, mlt, cmp1, cmp2 );
}

int main()
{
   int channel = Lib_get_video_video_channel();

   Lib_stop_freerun();

   // 取込モードをエンコーダに設定
   Lib_capt_set_xvideo_mode( ENCODER_XVIDEO_MODE, channel, -1 );

   // エンコーダの取込設定
   set_encoder();

   // 取込実行
  Lib_freeze( TRANSMIT );
}
```

留意事項　ありません。

Lib_capt_get_encoder

機　能　　エンコーダ設定値の取得

形　式

```
#include "f_video.h"
int  Lib_capt_get_encoder( int channel, int *start, int *mode, int *dir, int *signal,
                           int *mlt, int *cmp1, int *cmp2 );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　　エンコーダ設定値を取得します。

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

パラメータを取得する必要のない引数には、NULL を指定することができます。

① *channel* は設定するチャンネル番号です。

② ***start*** はエンコーダの開始方法です。

値	意	味
0	CPU	：画像入力の指示を受け次第、エンコーダのカウントを始めます。
1	外部トリガ	：画像入力の指示を受けると、外部トリガの入力を待ちます。外部トリガの入力を検出すると、エンコーダのカウントを始めます。
2	Z相	：画像入力の指示を受けると、Z相の入力を待ちます。Z相の入力を検出すると、エンコーダのカウントを始めます。

③ ***mode*** は 取込モードです。
本パラメータは、下記キャプチャボード使用時のみ有効になります。
下記ボード以外では、取得される値は不特定です。
- FHC3312
- FHC3313

値	意　味
0	エンコーダスキャンモード
1	エンコーダライン選択モード

④ ***dir*** はエンコーダの回転方向です。

値	意　味
0	CW
1	CCW

⑤ ***signal*** は検出するエンコーダの位相です。

値	意　　味
0	AB相
1	A相

⑥ ***mlt*** はエンコーダのサンプリングレートです。

値	意　　味
0	1倍
1	2倍
2	4倍　※***signal***にAB相を指定したときのみ使用できます。

⑦ ***cmp1*** は比較レジスタ１の値です。
　本パラメータに設定されたパルスのカウント数分だけ待機した後、取込を開始します。

⑧ ***cmp2*** は比較レジスタ２の値です。
　本パラメータに設定された値とパルスカウンタが一致した場合にライントリガを出力します。1ライン分のパルス数です。

戻り値

処理結果

値	定　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

エンコーダの各パラメータを取得します。

```
void get_encoder( int channel )
{
  int start, dir, signal, mlt, cmp1, cmp2;
  char buff[1024] = “¥0”;

  // パラメータの取得
  // 取得の必要がないパラメータには NULL を設定できます。
  Lib_capt_get_encoder( channel, start, NULL, dir, signal, mlt, cmp1, cmp2 );

  Lib_sprintf( buff, “%d,%d,%d,%d,%d”, start, dir, signal, mlt, cmp1, cmp2 );
  Lib_chrdisp( 1, 1, buff );
}
```

留意事項

ありません。

Lib_capt_set_strobe

機　能　ストロボの設定

形　式

```
#include "f_video.h"
int  Lib_capt_set_strobe( int channel, int enable, int expol,
                          int delay, int storobe_time );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　エンコーダを使用した取込設定をします。

本関数は下記キャプチャボードにて使用可能です。

- FVC06
- FHC3312
- FHC3313

① *channel* は設定するチャンネル番号です。

② ***enable*** はストロボの有効・無効設定です。

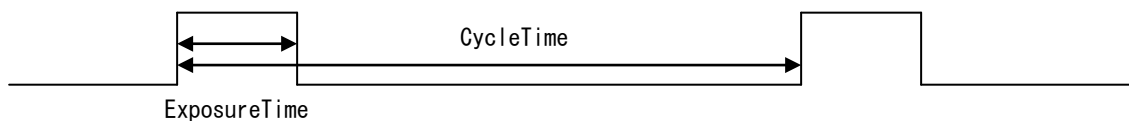
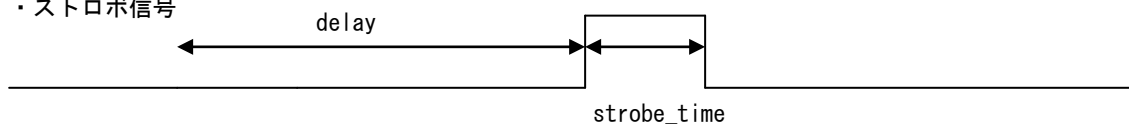
値	意　味
0	無効
1	有効

③ ***expol*** はストロボへ出力される信号の極性設定です。
FHC3312, FHC3313 では、ストロボへの出力は常に立ち下がり信号になります。

値	意　味
0	負（立ち下がり信号）
1	正（立ち上がり信号）

④ ***delayr*** は取込開始からのディレイ時間です。（ 単位：μsec ）
下図を参照ください。

⑤ ***storobbe_time*** はストロボ信号の立ち上がり（たち下がり）時間です。
（ 単位：μsec ）
下図を参照ください。

・取込信号

・ストロボ信号

【図：delay と storobe_time の関係】

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

ありません。

留意事項

ありません。

Lib_capt_get_line_counter

機　能

取込済みライン数の取得

形　式

```
#include "f_video.h"
int Lib_capt_get_line_counter( int iChannel, unsigned int *uipLineCnt );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたビデオ入力チャネルの取込済みライン数を取得します。
取込済みライン数とは、例えばラインセンサカメラで取込を行っている場合に、現在どのラインまで取込が完了しているかを示すものです。

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

① ***iChannel*** はビデオ入力チャネル番号です。

② ****uipLineCnt*** は取込済みライン数格納メモリへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

取込を行い、取込済みライン数を取得します。

```
int check_line_count( int channel )
{
    char buff[1024] = “¥0”;

  // 取込開始
  Lib_xfreeze();

  while(1)
  {
     unsigned int cnt = 0;

     // 取込完了確認
     if( FINISHED_TRANSMIT == Lib_get_video_status( channel ) )
        break;

        // 取込済みライン数の取得
        Lib_capt_get_line_counter( channel, &cnt );

        // 表示
        Lib_sprintf( buff, “%4d”, cnt );
      Lib_chrdisp( 1, 1, buff );


     // ループ文の最中では表示が更新されないため、ディレイを入れる。
        Lib_time_delay( 10 );
  }
  }
```

留意事項

ありません。

Lib_GetCaptureType

機　能

キャプチャタイプ取得

形　式

```
#include "f_video.h"
int  Lib_GetCaptureType( void );
```

901	902	903	904	FVL/LNX	高分解能
	○			○	

解　説

キャプチャタイプを取得します。

戻り値

キャプチャタイプ
以下のビット割付で搭載キャプチャボードのビットが１になります。

値	定　数	意　味
0x0001		CATS
0x0002		VCAP
0x0008	CAPT_TYPE_RICE	RICE001(a/b)
0x0010		PLUM
0x0010		FVC01
0x0040	CAPT_TYPE_FVC02	FVC02
0x0080	CAPT_TYPE_FVC05	FVC05
0x0100	CAPT_TYPE_FVC04	FVC04
0x0200	CAPT_TYPE_FVC06	FVC06
0x0400	CAPT_TYPE_GP440	FV-GP440
0x1000	CAPT_TYPE_FHC3312	FHC3312
0x2000	CAPT_TYPE_FHC3313	FHC3313

例

キャプチャタイプを取得し、ビデオ入力モードを設定します。

```
#include "f_video.h"

void set_video_input_mode( int  iChannel )
{
    if ( 0 != Lib_GetCaptureType() & 0x40  )
    {
        /* FVC02 */
        Lib_set_xvideo_input_mode( NORMAL_XVIDEO_MODE, iChannel, XC_55 );
    }
    else
    {
        Lib_set_video_input_mode( NORMAL_VIDEO_MODE, iChannel );
    }
}
```

留意事項

FVL/LNX

○本関数で取得するキャプチャタイプはチャンネル０に設定されているボードです。
特定のチャンネルのキャプチャタイプを取得する場合にはLib_GetCaptureType2関数を使用ください。

Lib_GetCaptureType2

機　能

キャプチャタイプ取得2

形　式

```
#include "f_video.h"
int  Lib_GetCaptureType2( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

特定のチャンネルのキャプチャタイプを取得します。

① ***channel*** はキャプチャタイプを調べたいチャンネル番号です。

戻り値

キャプチャタイプ
以下のビット割付で搭載キャプチャボードのビットが１になります。

値	定　　数	意　　味
0x0008	CAPT_TYPE_RICE	RICE001(a/b)
0x0040	CAPT_TYPE_FVC02	FVC02
0x0080	CAPT_TYPE_FVC05	FVC05
0x0100	CAPT_TYPE_FVC04	FVC04
0x0200	CAPT_TYPE_FVC06	FVC06
0x0400	CAPT_TYPE_GP440	FV-GP440
0x1000	CAPT_TYPE_FHC3312	FHC3312
0x2000	CAPT_TYPE_FHC3313	FHC3313

例

キャプチャタイプを取得し、ビデオ入力モードを設定します。

```
#include "f_video.h"

void set_video_input_mode( int  iChannel )
{
    if ( 0 != Lib_GetCaptureType2(0) & 0x40  )
    {
        /* FVC02 */
        Lib_set_xvideo_input_mode( NORMAL_XVIDEO_MODE, iChannel, XC_55 );
    }
    else
    {
        Lib_set_video_input_mode( NORMAL_VIDEO_MODE, iChannel );
    }
}
```

留意事項

ありません。

Lib_set_video_clock_rate

機　能

ビデオクロック設定

形　式

```
#include "f_video.h"
int Lib_set_video_clock_rate( int rate );
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△		

解　説

2倍速取込時にビデオクロックの設定を行います。

① ***rate*** は倍率です。

値	定　　数	意　　味
1		1倍(通常の取込)
2		2倍速取込

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-2		ハードウェア未対応。

例

2倍速カメラ用の設定を行います。

```
#include "f_video.h"

int set_camera_rate()
{
    return Lib_set_video_clock_rate(2);
}
```

留意事項

902

○本ライブラリは、FVC01 REV.2 以降で使用可能です。

903

○装置背面のシリアルナンバーNo.017299 までは、2 倍速取込は使用できません。

904

○本ライブラリは、FV904p, FV904pc のみで使用可能です。

Lib_get_video_clock_rate

機　能

ビデオクロック参照

形　式

```
#include "f_video.h"
int Lib_get_video_clock_rate(void);
```

901	902	903	904	FVL/LNX	高分解能
	△	○	△		

解　説

ビデオクロックの参照を行います。

戻り値

処理結果

値	定　　数	意　　味
1		1倍（通常の取込）
2		2倍速取込

例

現在のビデオクロックを参照します。

```
#include "f_video.h"

int get_camera_rate()
{
    return Lib_get_video_clock_rate();
}
```

留意事項

902

○本ライブラリは、FVC01 REV.2 以降で使用可能です。

903

○装置背面のシリアルナンバーNo.017299 までは、1 倍速取込のみです。

904

○本ライブラリは、FV904p, FV904pc のみで使用可能です。

Lib_capt_grab_start

機　能

連続取込の開始

形　式

```
#include "f_video.h"
int Lib_capt_grab_start( int mem_no[], int frame_su, unsigned int plane );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

カレントビデオ入力チャネルの連続画像を、mem_no[0]で指定される画像メモリを先頭として順番に取込ます。mem_no[frame_su － 1]まで取り込むとmem_no[0]に戻って取込動作を継続します。
連続取込動作を停止する場合は、後述のLib_capt_grab_end()をコールして下さい。

① ***mem_no[]*** は取込先画像メモリ番号配列です。

② ***frame_su*** は画像メモリ番号配列の要素数です。

③ ***plane*** は画像入力プレーンです。
注：現在のところ GRAY_PLANE しか選択できません。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

画像取込と処理を並列実行します。

```
void func( void )
{
        int     x, y;
        int     mem_no[2];
        int     index;

        mem_no[0] = Lib_alloc_gray_memory();
        mem_no[1] = Lib_alloc_gray_memory();
        if( NORMAL_RETURN == Lib_capt_grab_start( mem_no, 2, GRAY_PLANE ) )
{
            while(1)
            {
                Lib_capt_wait_event( Lib_get_video_channel(), 0 );
                Lib_capt_get_frame_index( Lib_get_video_channel(), &index );
                index -= 1;     if( index < 0 )  index = 0;

                /* 何らかの処理 */

                if( 0 != Lib_see_current_position_real(&x,&y) )
                        break;
            }
            Lib_capt_grab_end( Lib_get_video_channel() );
        }
}
```

留意事項

○現在のところ入力プレーンは、濃淡プレーンのみに対応しております。

FVL/LNX

○FVC06, FHC3312, FHC3313 をランダムトリガモードで使用する場合には、1 フレーム辺りの取り込み時間がLib_capt_set_reg関数のFVC06_TRIGGER_CYCLETEIME(FVC06の場合)パラメータで指定した値になります。

例：　/*　1 フレーム 50ms で連続取り込みを行う場合（ランダムトリガモード）　*/

```
void capt_grab_fvc06( void )
{
        int channel = Lib_get_video_channel();

        // 連続取り込みの 1 フレーム辺りの時間
        const int cycle_time = 50 * 1000; // 50ms

        const int cnts = 16;
        int mem_no[cnts];
        int i;

        // メモリの確保
        for( i = 0; i < cnts; i++ )
          mem_no[i] = Lib_alloc_gray_memory();

        // 連続取り込みの 1 フレーム辺りの取り込み時間を設定する。
        Lib_capt_set_reg( channel, FVC06_TRIGGER_CYCLETIME, cycle_time );
```

```
        // 連続取り込み開始
        Lib_capt_grab_start( mem_no[0], cnts, GRAY_PLANE );
  for( ;; )
  {
    int index = 0;
    Lib_capt_wait_event( channel, 0 );
    Lib_capt_get_frame_index( channel, &index );
    index -= 1; if( index < 0 )  index = 0;

    /* 何らかの処理 */

    if( 0 != Lib_see_current_position_real(&x,&y) )
      break;
  }
  Lib_capt_grab_end( channel );
}
```

Lib_capt_get_frame_index

機　能

フレームインデックス番号の取得

形　式

```
#include "f_video.h"
int Lib_capt_get_frame_index( int channel, unsigned int *frame_index );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

指定されたビデオ入力チャネルのフレームインデックス番号を取得します。
フレームインデックス番号とは、連続取込が現在どこまで進んでいるかを示すもので、先頭の画像メモリに映像データを転送中である場合は0，2番目の画像メモリに映像データを転送中である場合は1，3番目に転送中であれば2のように変化していきます。

① ***channel*** ***は連続取込中の***ビデオ入力チャネル***番号です。***

② ****frame_index*** はフレームインデックス番号格納***メモリへのポインタです。***

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

Lib_capt_grab_start()の例を参照してください。

留意事項

ありません。

Lib_capt_wait_event

機　能

画像入力イベントの待機

形　式

```
#include "f_video.h"
int Lib_capt_wait_event( int channel, int wait_event );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

指定されたビデオ入力チャネルの連続取込が、待機イベントで指定される状態になるまで待機します。

① ***channel*** は連続取込中のビデオ入力チャネル番号です。

② ***wait_event*** は待機イベントです。

値	定　　数	意　　味
0		画像入力終了まで待機します。
1		露光完了まで待機します。
2		露光完了まで待機します。

注：FVC02, FVC05, FVC06, FV-GP440, FHC3312, FHC3313では1（露光完了まで待機）は使用できません。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

Lib_capt_grab_start()の例を参照してください。

留意事項

○FVC02, FVC05, FVC06, FV-GP440, FHC3312, FHC3313をご使用の場合は、待機イベントに1(露光完了まで待機)は指定出来ません。
値2(露光完了まで待機)は、FVC04のみ使用可能です。

Lib_capt_grab_end

機　能

連続取込の停止

形　式

```
#include "f_video.h"
int Lib_capt_grab_end( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

指定されたビデオ入力チャネルの連続取込を停止します。

① ***channel*** は連続取込中のビデオ入力チャネル番号です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

Lib_capt_grab_start()の例を参照してください。

留意事項

ありません。

Lib_capt_set_frame_number

機　能

連続取込の取り込み枚数指定

形　式

#include "f_video.h"
int Lib_capt_set_frame_number(int ***channel,*** unsigned int ***frame_num***);

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたビデオ入力チャネルの連続取込の取り込み枚数を指定します。

① ***channel*** は連続取込中のビデオ入力チャネル番号です。
② ***frame_num***は連続取込で取り込むフレーム枚数です。
　0を指定すると無限枚数取り込み実行になります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

```
/* 取り込み枚数を指定して連続取り込みを実行します */

void capt_grab_fvc06( void )
{
   int chennal = Lib_get_video_channel();
   const int cnts = 16; // 取り込み枚数
   int mem_no[cnts];     // メモリ配列

          // メモリの確保
          for( i = 0; i < cnts; i++ )
     mem_no[i] = Lib_alloc_gray_memory();

   /* 連続取り込みの取り込み枚数の指定 */
   Lib_capt_set_frame_number( channel, cnts );

   // 連続取り込み開始
          if( NORMAL_RETURN == Lib_capt_grab_start( mem_no, cnts, GRAY_PLANE ) )
       {
               while(1)
               {
                   int index;

                   Lib_capt_wait_event( channel, 0 );
                   Lib_capt_get_frame_index( channel, &index );
                   index -= 1;     if( index < 0 )  index = cnts-1;

                   /* 何らかの処理 */
```

```
            if( 0 != Lib_see_current_position_real(&x,&y) )
                break;
        }
        Lib_capt_grab_end( Lib_get_video_channel() );
    }


}
```

留意事項

○本関数はFVC06, FV-GP440, FHC3312, FHC3313で使用可能です。
○本関数を使用して取込枚数を指定した連続取り込みを行う場合には、Lib_capt_grab_start関数の前に必ず本関数を一度コールしてください。
○本関数を使用して取込枚数を指定した後に、取込を無限実行にする場合には、frame_numに0を入力してください。

Lib_capt_set_reg

機　能

取り込みパラメータの設定

形　式

```
#include "f_video.h"
int Lib_capt_set_reg( int channel, int regID, int value );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたビデオ入力チャネルの指定されたパラメータを指定値で設定します。

① ***channel*** はパラメータを設定するビデオ入力チャネル番号です。
② ***regID*** は設定するパラメータのIDです。
③ ***value*** は指定したパラメータIDに設定する値です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

```
/* FVC06 を使用したラインセンサ取り込みで、エンコーダパラメータの変更をおこないます */

void grab_fvc06_lsensor( void )
{
  int channel = Lib_get_video_channel();

  int enc_signale  = 0;  // エンコーダの位相設定               : A 相
    int enc_sampling = 0;  // エンコーダ信号のサンプリング速度 : 1 倍
  int enc_start    = 1;  // エンコーダの開始方法               : 外部トリガ
  int enc_cmp1     = 0;  // エンコーダ信号からの取り込み遅延カウント数
  int enc_cmp2     = 30; // エンコーダ信号からの取り込み周期[ line/pulse ]

   /* 取り込みモードをエンコーダに変更します */
  Lib_set_xvideo_input_mode( ENCODER_XVIDEO_MODE, channel, -1 /* not use */ );

  /* エンコーダのパラメータを設定します。 */
  // エンコーダの位相
  Lib_capt_set_reg( cahnnel, FVC06_ENCODER_ENCSIGNAL, enc_signal );

  // サンプリング速度
  Lib_capt_set_reg( cahnnel, FVC06_ENCODER_ENCMLT, enc_sampling );

  // エンコーダの開始方法
  Lib_capt_set_reg( cahnnel, FVC06_ENCODER_ENCSTART, enc_start );

  // 比較レジスタ 1
```

```
  Lib_capt_set_reg( cahnnel, FVC06_ENCODER_COMPREG1, enc_cmp1 );

  // 比較レジスタ２
  Lib_capt_set_reg( cahnnel, FVC06_ENCODER_COMPREG2, enc_cmp2 );

  /*  取り込み */
  Lib_freeze( TRANSMIT );
}
```

例2：

```
/* FVC06を使用したラインセンサ取り込みで、スキャンレートの変更をおこないます */
void grab_fvc06_lsensor2( void )
{
  int channel = Lib_get_video_channel();

  int cycle_time   = 300;  // スキャンレート[us]

   /* 取り込みモードをランダムトリガに変更します */
  Lib_set_xvideo_input_mode( RANDOM_XVIDEO_MODE, channel, -1 /* not use */ );

  /* スキャンレートを設定します。 */
  Lib_capt_set_reg( cahnnel, FVC06_TRIGGER_CYCLETIME, cycletime );

  /*  取り込み */
  Lib_freeze( TRANSMIT );
 }
```

留意事項

〇FVC06を使用する場合のパラメータIDについての説明は、巻末の『3. Lib_capt_set_reg 補足説明』を参照ください。

〇本関数は、下記のキャプチャボードで使用可能です。
- FVC06
- FHC3312
- FHC3313

Lib_capt_get_reg

機　能

取り込みパラメータの取得

形　式

#include "f_video.h"
int Lib_capt_get_reg(int ***channel,*** int ***regID,*** int ****value***);

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたビデオ入力チャネルの指定されたパラメータの現在の設定値を取得します。

① ***channel*** はパラメータを設定するビデオ入力チャネル番号です。
② ***regID*** は設定するパラメータのIDです。
③ ****value*** は指定したパラメータIDの設定値を取得するメモリへのポインタです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

```
/* FVC06 で設定済みのエンコーダパラメータを取得します */

void grab_fvc06_lsensor( void )
{
  int channel = Lib_get_video_channel();

  int enc_signale;  // エンコーダの位相設定
    int enc_sampling; // エンコーダ信号のサンプリング速度
  int enc_start;    // エンコーダの開始方法
  int enc_cmp1;     // エンコーダ信号からの取り込み遅延カウント数
  int enc_cmp2;     // エンコーダ信号からの取り込み周期[ line/pulse ]

  /* エンコーダのパラメータを取得します。 */
  // エンコーダの位相
  Lib_capt_get_reg( cahnnel, FVC06_ENCODER_ENCSIGNAL, &enc_signal );

  // サンプリング速度
  Lib_capt_get_reg( cahnnel, FVC06_ENCODER_ENCMLT, &enc_sampling );

  // エンコーダの開始方法
  Lib_capt_get_reg( cahnnel, FVC06_ENCODER_ENCSTART, &enc_start );

  // 比較レジスタ 1
  Lib_capt_get_reg( cahnnel, FVC06_ENCODER_COMPREG1, &enc_cmp1 );

  // 比較レジスタ 2
```

```
  Lib_capt_get_reg( cahnnel, FVC06_ENCODER_COMPREG2, &enc_cmp2 );
}
```

留意事項

〇FVC06を使用する場合のパラメータIDについての説明は、巻末の『3. Lib_capt_set_reg 補足説明』を参照ください。

〇本関数は、下記のキャプチャボードで使用可能です。
- FVC06
- FHC3312
- FHC3313

Lib_capt_send_cmd

機　能　カメラリンク経由のシリアルコマンド送信

形　式

```
#include "f_video.h"
int Lib_capt_send_cmd( int channel, char *send, char *recv );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

カメラに対し、カメラリンク経由でシリアル通信を行います。
カメラ(メーカー)固有のSTX/ETXはライブラリ内で自動的に付加されるため、
送信コマンド列に加える必要はありません。
(※また、日立国際電気製カメラの場合には、checksumデータも併せてライブラリ内部で
付加します)

① ***channel*** はコマンドを送信するチャネル番号です。
② *send は送信するコマンド列です。
関数内部でカメラ (メーカー) 固有のSTX/ETXを自動的に付加します。
送信すコマンドには、STX/ETXを除いたコマンド列を入力ください。
③ *recv はコマンド送信後に、カメラから受信したコマンド列です。
関数内部で受信コマンドに合わせた配列サイズのチェックは行いません。
受信するコマンド長に対し十分に足りる配列サイズを確保してください。

戻り値　処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

① KP-F200SCLに対し、シャッタスピードをPRESET4に変更するコマンドを送ります。

```
void change_shutter_spped_kpfxxxx( int channel )
{
        char send[] = "01FF0108040000";
        chae recv[32];
        Lib_send_cmd( channel, send, recv );
        Lib_printf( recv );
}
```

② KP-F200SCL に対し、現在設定されているシャッタースピードを取得するコマンドを送ります。

```
void change_shutter_spped_kpfxxxx( int channel )
{
        char send[] = "00FF8108000000";
        chae recv[32];
        Lib_send_cmd( channel, send, recv );
        Lib_printf( recv );
}
```

サンプルとして、sample/sample_send_cmd 以下にサンプルソースがインストールされています。

留意事項　本関数はFVC06, FHC3312, FHC3313専用です。

Lib_show_serial_console

機　能

カメラリンク経由のシリアルコマンド送信用windowの表示

形　式

```
#include "f_video.h"
int Lib_show_send_cmd( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カメラに対しカメラリンク経由でシリアル通信を行うためのwindowを表示します。
カメラ(メーカー)固有のSTX/ETXはライブラリ内で自動的に付加されるため、
送信コマンド列に加える必要はありません。
(※また、日立国際電気製カメラの場合には、checksumデータも併せてライブラリ内部で
付加します　)

本関数は下記キャプチャボードにて使用可能です。
- FVC06
- FHC3312
- FHC3313

本関数にて表示されるwindowの機能を使用する場合には、お使いの画像処理装置にキーボードを接続する必要があります。

① ***channel*** はコマンドを送信するチャネル番号です。

戻り値

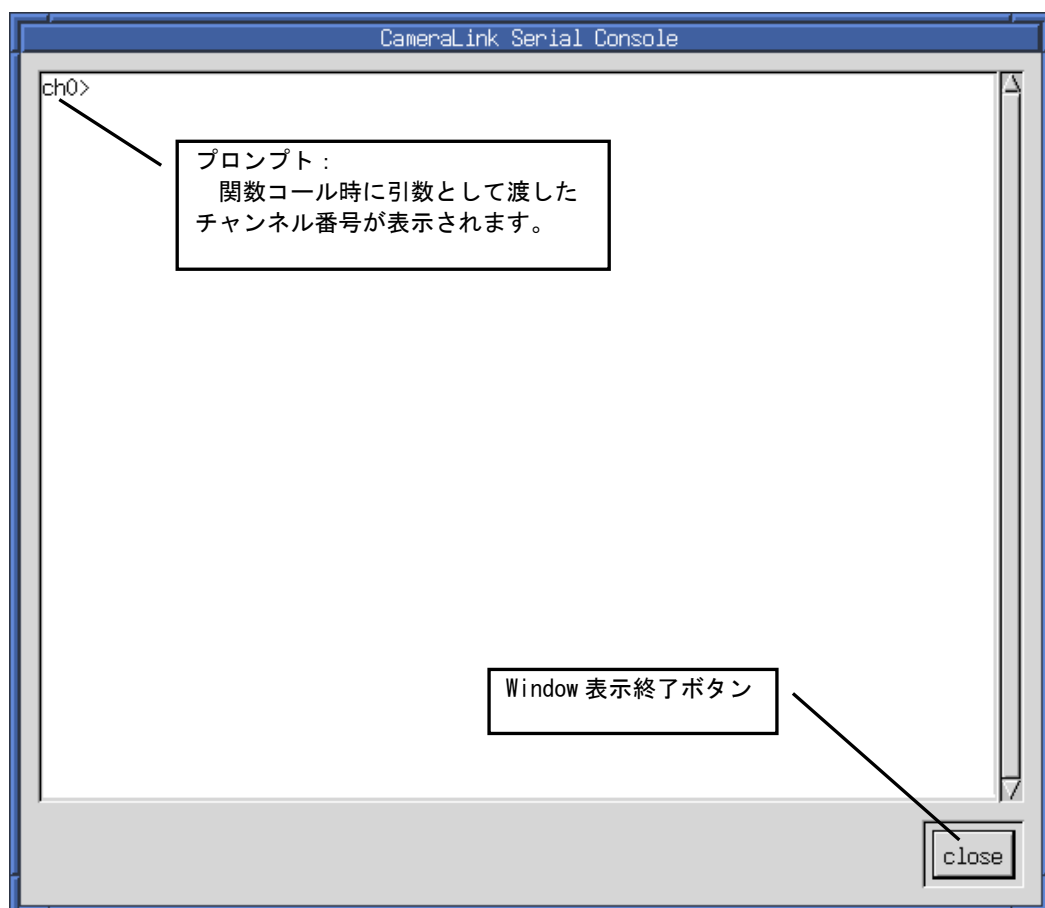
処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

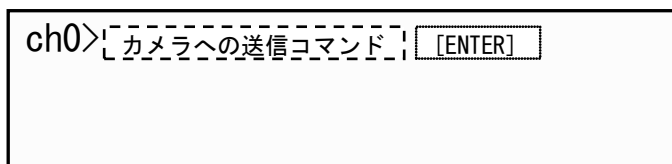
ありません。

留意事項

□windowの操作説明

・コンソールの使用方法

プロンプト（チャンネル０に対して使用する場合には"Ch0>"と表示されます）以降に、カメラに送信したいコマンド文字列を入力してください。

[Enter]キーを入力することにより、入力されたコマンド文字列がカメラに送信されます。

ch0>カメラへの送信コマンド [ENTER]

Lib_display_ccfg_menu

機　能

カメラ調整用ツールの表示

形　式

```
#include "f_video.h"
int Lib_display_ccfg_menu( int lang );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カメラ調整用ツールを表示します。

本関数にて表示されるツールの機能を使用する場合には、お使いの画像処理装置にキーボードを接続する必要があります。

① ***lang*** はメニュー表示の言語設定です。

値	意　味
0	英語
1	日本語

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

ありません。

留意事項

- ○ 本関数実行中は、フリーラン動作が停止されます。また、関数から復帰後もフリーランを開始しません。
- ○ 本関数実行中に取込サイズを変更した場合、関数から復帰後のカレントメモリのサイズも変更されます。
- ○ 本関数実行中に変更された取込のパラメータ（ビデオ入力モード等）は関数から復帰後も継続されます。
- ○ カメラ調整用ツールの画面、使用方法については、FVL/LNX 操作説明書を参照ください。

Lib_set_transmit_stpos

機　能

映像出力始点座標の設定

形　式

```
#include "f_video.h"
int Lib_set_transmit_stpos( int xs, int ys );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

映像を出力する際のビデオウィンドウ上の始点座標を設定します。

① *xs* は映像出力始点X座標です。

② *ys* は映像出力始点Y座標です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

Lib_get_transmit_stpos()の例をご覧ください。

留意事項

ありません。

Lib_get_transmit_stpos

機　能

映像出力始点座標の取得

形　式

```
#include "f_video.h"
int Lib_get_transmit_stpos( int *xs, int *ys );
```

901	902	903	904	FVL/LNX	高分解能
				○	○

解　説

映像を出力する際のビデオウィンドウ上の始点座標を取得します。

① ***xs** は映像出力始点X座標です。

② ***ys** は映像出力始点Y座標です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました
-1	ERROR_RETURN	異常終了しました

例

現在の映像出力位置からX方向に+10移動した位置に濃淡画像を出力します。

```
void transmit( void )
{
        int x, y;
        Lib_get_transmit_stpos( &x, &y );
        x += 10;
        Lib_set_transmit_stpos( x, y );
        Lib_video_transmit( GRAY_PLANE );
}
```

留意事項

ありません。

Lib_cl_sio_open

機　能

カメラ通信オープン

形　式

```
#include "f_cl.h"
int Lib_cl_sio_open( int channel, int timeout );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号のカメラリンク通信をオープンします。

① ***channel***はチャネル番号です。

② ***meout***は通信タイムアウト値(msec)です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。

Lib_cl_sio_close

機　能

カメラ通信クローズ

形　式

```
#include "f_cl.h"
int Lib_cl_sio_close( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号のカメラリンク通信をオープンします。

① ***channel***はチャネル番号です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。

Lib_cl_setmode

機　能

取込モードの設定

形　式

```
#include "f_cl.h"
int Lib_cl_setmode( int channel, int mode, int sw );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号の画像取込モードを設定します。

① ***channel***はチャネル番号です。

② ***mode***は設定するモード番号です。

定　　数	意　　味
CLMODE_WOI	WOIモードを設定します。
CLMODE_PARTIAL	パーシャルスキャンモードを設定します。
CLMODE_LDMA	LATE DMAモードを設定します。

③ ***sw***はモードを設定します。

値	定　　数	意　　味
0	ON	有効
1	OFF	無効

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。
○部分読み出しを実行する際は、必ずLib_cl_setvpart関数にて垂直取込の設定を行ってから、Lib_cl_setmode関数にてパーシャルスキャンモードに移行してください。

Lib_cl_getmode

機　能

取込モードの取得

形　式

```
#include "f_cl.h"
int Lib_cl_getmode( int channel, int mode, int *sw );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号の画像取込モードを取得します。

① ***channel***はチャネル番号です。

② ***mode***は取得するモード番号です。

定　　数	意　　味
CLMODE_WOI	WOIモードを取得します。
CLMODE_PARTIAL	パーシャルスキャンモードを取得します。
CLMODE_LDMA	LATE DMAモードを取得します。
CLMODE_CAMTYPE	カメラ種別を取得します。

③ ***sw***は取得先です。

値	定　　数	意　　味
0	ON	有効
1	OFF	無効

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。

Lib_cl_setvpart

機　能

画像取込垂直ライン設定

形　式

```
#include "f_cl.h"
int Lib_cl_setvpart( int channel, int vstart, int lines, int vofs );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号の垂直画像取込設定を行います。

① ***channel***はチャネル番号です。

② ***vstart***は取込開始ラインです。

③ ***lines***は取込ライン数です。

④ ***vofs***は取込オフセット量です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。
○設定したパラメータはパーシャルスキャンモード時のみ有効です。
○部分読み出しを実行する際は、必ずLib_cl_setvpart関数にて垂直取込の設定を行ってから、Lib_cl_setmode関数にてパーシャルスキャンモードに移行してください。

Lib_cl_force_trigger

機　能

内部トリガ発行

形　式

```
#include "f_cl.h"
int Lib_cl_force_trigger( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネル番号に内部トリガを発行します。

① ***annel***はチャネル番号です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04専用です。

Lib_cl_csb_shutter_speed

機　能

露光時間設定

形　式

```
#include "f_cl.h"
int Lib_cl_csb_shutter_speed( int channel, int value );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カメラ側で使用される露光時間を設定します。

①　***channel*** はチャネル番号です。

②　***value*** は露光時間を設定します。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_ssr

機　能

リセット後の露光時間設定

形　式

```
#include "f_cl.h"
int Lib_cl_csb_ssr( int channel, int number, int value );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

マルチスロープ機能におけるリセット後の露光時間を設定します。

① ***annel*** はチャネル番号です。

② ***number*** は設定するマルチスロープ番号です。(0～2)

③ ***value*** はリセット後の露光時間を設定します。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_setreg

機　能

カメラ設定書き込み

形　式

```
#include "f_cl.h"
int Lib_cl_csb_setreg( int channel, int regno, int data );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネルのカメラ設定データを書き込みます。

① ***hannel***はチャネル番号です。

② ***regno***は設定データ番号です。

③ ***data***は書き込むデータです。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_getreg

機　能

カメラ設定読み出し

形　式

```
#include "f_cl.h"
int Lib_cl_csb_getreg( int channel, int regno, int *data );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネルのカメラ設定データを読み出します。

① ***hannel***はチャネル番号です。

② ***regno***は設定データ番号です。

③ ***data***は読み出しデータ取得先です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_setwindow

機　能

ウィンドウデータ設定

形　式

```
#include "f_cl.h"
int Lib_cl_csb_setwindow(
        int channel, int number, int sw, int sx, int sy, int dx, int dy );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

WOI(Window Of Interest)機能におけるウィンドウデータをカメラに設定します。

① ***hannel***はチャネル番号です。

② ***number***はウィンドウ番号です。(0～15)

③ ***sw***は有効(1)、無効(0)の切り換えです。

④ ***sx***, ***sy***はウィンドウ左上のx座標及びy座標です。

⑤ ***dx***, ***dy***はそれぞれx方向、y方向のウィンドウサイズです。

※ ***sx***, ***dx***は必ず偶数、***dx***は最小40となるようにして下さい。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリはFVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。
○WOIウィンドウの設定は、以下の条件を満たす必要があります。

・始点X	32の倍数
・始点Y	2の倍数
・横方向サイズ	32の倍数、64画素以上
・縦方向サイズ	2の倍数

Lib_cl_csb_getwindow

機　能

ウィンドウデータ取得

形　式

```
#include "f_cl.h"
int Lib_cl_csb_getwindow(
        int channel, int number, int *sw, int *sx, int *sy, int *dx, int *dy );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

WOI(Window Of Interest)機能におけるウィンドウデータをカメラから取得します。

① ***channel***はチャネル番号です。

② ***number***はウィンドウ番号です。(0～15)

③ ****sw***は有効(1)、無効(0)の取得先です。

④ ****sx***, ****sy***はウィンドウ左上のx座標及びy座標の取得先です。

⑤ ****dx***, ****dy***はそれぞれx方向、y方向のウィンドウサイズ取得先です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_setup_woi

機　能

WOIセットアップ

形　式

```
#include "f_cl.h"
int Lib_cl_csb_setup_woi( int channel );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カメラに登録されているウィンドウ設定でWOIを登録します。

① ***channel***はチャネル番号です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_csb_setup_woi_local

機　能

ローカルデータによるWOIセットアップ

形　式

#include "f_cl.h"
int Lib_cl_csb_setup_woi_local(int ***channel***, int ***data**, int ***wcount***);

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定ウィンドウデータにより、WOIを登録します。

① ***channel***はチャネル番号です。
② ***data***はウィンドウデータです。

0	有効(1), 無効(0)	
1	sx	sy
2	dx	dy
・・・	・・・	
(wcount-1)x3	有効(1), 無効(0)	
(wcount-1)x3+1	sx	sy
(wcount-1)x3+2	dx	dy

③ ***wcount***は登録するウィンドウデータ数です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝テリー製CSBシリーズ専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。
○WOIウィンドウの設定は、以下の条件を満たす必要があります。

・始点X　　32の倍数
・始点Y　　2の倍数
・横方向サイズ　　32の倍数、64画素以上
・縦方向サイズ　　2の倍数

Lib_cl_kpf_write

機　能　　カメラ設定書き込み

形　式

```
#include "f_cl.h"
int Lib_cl_kpf_write( int channel, int cmd, int data );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　　指定したチャネルのカメラ設定データを書き込みます。

① ***channel***はチャネル番号です。

② ***cmd***は設定データ番号です。

③ ***data***は書き込むデータです。

～ KP-F100BCL ～

cmd	意　味	data	機能
KPF100_FD	フレームオンデマンド	0	有効
		1	無効
KPF_MODE	動作モード	0	ノーマルモード
		1	2トリガ
		2	1トリガ
KPF_SHTSPD	シャッタスピード	0	1/30
		1	1/125
		2	1/250
		3	1/1000
		4	1/2000
		5	1/4000
		6	1/10000
		7	1/50000
KPF100_RGAIN	ラフゲイン	0	0dB
		1	6dB
		2	12dB
KPF100_GAIN	ゲイン	0～50	微調整値
KPF_BLACKLV	ブラックレベル	0～50	オフセット設定値
KPF100_4x	4倍速(設定禁止)	0	ON(設定禁止)
		1	OFF
KPF100_TRIG	トリガ	0	通常(NORMAL)
		1	反転(INVERT)

～ KP-F120CL ～

cmd	意　　味	data	機能
KPF_MODE	動作モード	0	ノーマルモード
		1	1トリガ
		2	2トリガ
		3	4倍速
		4	4倍速+1トリガ
		5	スミア+1トリガ
		6	スミア+2トリガ
		7	スミア+4倍速+1トリガ
		8	ノーマルシャッタ
		9	固定シャッタ
		10	スミア+固定シャッタ
		11	パーシャルスキャン+中央
		12	パーシャルスキャン+上部
		13	パーシャル+中央+1トリガ
		14	パーシャル+上部+1トリガ
KPF_SHTSPD	シャッタスピード	0	1/30
		1	1/60
		2	1/125
		3	1/250
		4	1/1000
		5	1/2000
		6	1/10000
		7	1/50000
KPF120_GAIN	ゲイン	0～15	ゲイン設定値
KPF_BLACKLV	ブラックレベル	0～50	オフセット設定値
KPF120_AREA	パーシャルスキャンエリア	0	16ライン
		1	32ライン
		2	64ライン
		3	128ライン
		4	256ライン
		5	512ライン

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と日立国際電気製KP-F100BCL, KP-F120CL専用です。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_kpf_read

機　能

カメラ設定の読み出し

形　式

```
#include "f_cl.h"
int Lib_cl_kpf_read( int channel, int cmd );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネルのカメラ設定データを読み出します。

① ***channel***はチャネル番号です。

② ***cmd***は設定データ番号です。

戻り値

処理結果

値	定　数	意　味
>=0		カメラパラメータ
-1	ERROR_RETURN	異常終了しました。

※ パラメータの詳細はLib_cl_kpf_writeを参照してください。

例

留意事項

○本ライブラリは、FVC04と日立国際電気製KP-F100BCL，KP-F120CL専用です。
○必ずLib_cl_sio_openにてカメラ通信をオープンしてから使用してください。
○終了後はLib_cl_sio_closeにてカメラ通信をクローズしてください。

Lib_cl_ik_setmode

機　能

カメラ動作モードの設定

形　式

```
#include "f_cl.h"
int Lib_cl_ik_setmode( int channel, int mode );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定したチャネルのカメラ設定データを読み出します。

① ***channel***はチャネル番号です。

② ***mode***は設定する部分読み出しモードです。

mode	意　　　味
IKMODE_FULL	ノーマルモード
IKMODE_700	画面垂直方向中心700ライン取込
IKMODE_560	画面垂直方向中心560ライン取込
IKMODE_400	画面垂直方向中心400ライン取込
IKMODE_70	画面垂直方向中心70ライン取込

戻り値

処理結果

値	定　　数	意　　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

留意事項

○本ライブラリは、FVC04と東芝製IK-SX1専用です。
○部分読み出しを実行する際は、必ずLib_cl_setvpart関数にて垂直取込の設定を行ってから、Lib_cl_setmode関数にてパーシャルスキャンモードに移行してください。

Lib_fvcm01_gpio_input

機　能

GPIO(汎用入出力)入力状態読み出し

形　式

```
#include "f_m01.h"
int Lib_fvcm01_gpio_input(int bno, int *bit_pattern);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたキャプチャボード番号のGPIO(汎用入出力)ポートの入力状態を読み出します。

① ***bno***はボード番号です。必ず0を指定して下さい。

② ****bit_pattern***は読み出した入力ビットパタンです。

ビット位置	機　　能
32～4	無効
3	GPI 3 の入力状態
2	GPI 2 の入力状態
1	GPI 1 の入力状態
0	GPI 0 の入力状態

※ ビット=0の時OFF、ビット=1の時ONとなります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_m01.h"

/* 指定された入力ポートの ON/OFF 状態を取得します。 */
int read_input_level(int pos)
{
    int d;
    d = Lib_fvcm01_gpio_input(0, &d);
    if(d < 0)
        return d;
    return (d >> pos) & 1;

}
```

留意事項

○本ライブラリは、FV1100-LNX専用です。

Lib_fvcm01_gpio_output

機　能　GPIO(汎用入出力)出力

形　式

```
#include "f_m01.h"
int Lib_fvcm01_gpio_output(int bno, int bit_pattern);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたキャプチャボード番号のGPIO(汎用入出力)ポートに出力します。

① ***bno***はボード番号です。必ず0を指定して下さい。

② ***bit_pattern***は出力するビットパタンです。

ビット位置	機　　　能
32～4	無効
3	GPI 3 への出力
2	GPI 2 への出力
1	GPI 1 への出力
0	GPI 0 への出力

※ ビット=0の時OFF、ビット=1の時ONとなります。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_m01.h"

/*指定されたポートに ON/OFF 出力します。 */
int read_input_level(int pos, int data)
{
    int d, ret;
    if(data)
            d = Lib_fvcm01_gpio_input(0, data) | (1<<pos);
    else
            d = Lib_fvcm01_gpio_input(0, data) & ~(1<<pos);
    return Lib_fvcm01_gpio_output(0, d);

}
```

留意事項

○本ライブラリは、FV1100-LNX専用です。
○出力ポートが汎用出力モードに設定されている時のみ有効です。

Lib_fvcm01_set_conf

機　能　　キャプチャボード設定書込

形　式

```
#include "f_m01.h"
int FVLCALL Lib_fvcm01_set_conf(int bno, int kind, int flag);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説　　指定されたキャプチャボード番号の各種設定を書き込みます。

① ***bno***はボード番号です。必ず0を指定して下さい。

② ***kind***は書き込む設定項目です。

定　　数	意　　味
FVCM01_CONFIG_LOGIC	トリガ入力論理
FVCM01_CONFIG_IN	入力ポート設定
FVCM01_CONFIG_OUT	出力ポート設定

③ ***flag***は設定する値です。設定可能な値は以下のようになります。

トリガ入力論理設定(kind = FVCM01_CONFIG_LOGICの時、マスク設定)

定　　数	意　　味
FVCM01_TRG_POS_LOGIC	トリガ入力モード正論理設定
FVCM01_TRG_NEG_LOGIC	トリガ入力モード負論理設定
FVCM01_INDEX_POS_LOGIC	INDEX入力モード正論理設定
FVCM01_INDEX_NEG_LOGIC	INDEX入力モード負論理設定

入力ポート設定(kind = FVCM01_CONFIG_INの時、選択設定)

定　　数	意　　味
FVCM01_TRGIN_SEPERATE	トリガ入力独立モード
FVCM01_TRGIN_ALL	トリガ入力ORモード
FVCM01_GPI_MODE	汎用GPIモード

出力ポート設定(kind = FVCM01_CONFIG_OUTの時、選択設定)

定　　数	意　　味
FVCM01_TRGOUT_SEPERATE	トリガ出力独立モード
FVCM01_TRGOUT_ALL	トリガ出力ORモード
FVCM01_ GPO_MODE	汎用GPOモード

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_m01.h"

/* 各入力端子を OR して、全チャネルにトリガを入力する設定です。 */
int set_trgin_from_all(void)
{
    return Lib_fvcm01_set_conf(0, FVCM01_CONFIG_IN, FVCM01_TRGIN_ALL);
}
/* トリガ入力論理を反転します。 */
int set_trgin_logic(int mode)
{
    return Lib_fvcm01_set_conf(0, FVCM01_CONFIG_LOGIC, FVCM01_TRG_NEG_LOGIC);
}
```

留意事項

○本ライブラリは、FV1100-LNX専用です。

Lib_fvcm01_get_conf

機　能

キャプチャボード設定読出

形　式

```
#include "f_m01.h"
int FVLCALL Lib_fvcm01_get_conf(int bno, int kind);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

指定されたキャプチャボード番号の各種設定を読み出します。

① ***bno***はボード番号です。必ず0を指定して下さい。

② ***kind***は読み出す設定項目です。
　※ 詳細はLib_fvcm01_set_conf()をご参照下さい。

戻り値

処理結果

値	定　　数	意　　味
>0		各設定項目値（Lib_fvcm01_set_confを参照して下さい。）
-1	ERROR_RETURN	異常終了しました。

例

Lib_fvcm01_set_cnof()をご参照下さい。

留意事項

○本ライブラリは、FV1100-LNX専用です。

Lib_fie_freeze

機　能

FIE画像オブジェクトへの画像取込

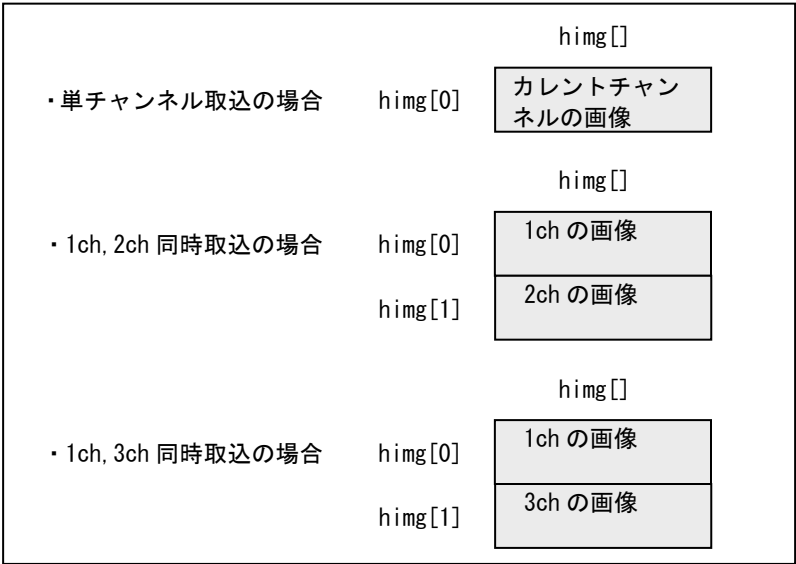
形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_freeze(FHANDLE *himg, INT num);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

画像取込指示を発行し、カメラから画像を入力します。
関数は取込が完了するまで制御を返しません。

① ***himg*** は取込先の FIE 画像オブジェクトを格納した配列の先頭アドレスです。

取り込まれた画像は、配列の先頭インデックスから若い番号順に格納されます。

取込先のFIE画像オブジェクトは、下記の条件を満たしている必要があります。
- 画像タイプがF_IMG_UC8
- チャネル数が1
- 画像サイズが取込サイズと等しいこと

また、指定するFIE画像オブジェクトはルート画像でなくてはいけません。

② ***num***は****himg***で指定した取込先FIE画像オブジェクトの配列個数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

```
#include "f_video.h"
#include "f_stdlib.h"
#include "f_fie.h"
#include "f_fieutil.h"

/* カレントチャネルの取込実行 */
void freeze( void )
{
    FHANDLE himg = NULL;

    // 取込サイズ取得
    INT fx = Lib_get_fx_size();
    INT fy = Lib-get_fy_size();

    // 取込先画像オブジェクトの確保
    himg = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );

    // 取込実行
    Lib_fie_freeze( &himg, 1 );
}

/* 2ch 同時取込実行 */
void 2ch_freeze( void )
{
    FHANDLE himg[2] = { NULL, NULL };

    // 取込サイズ取得
    INT fx = Lib_get_fx_size();
    INT fy = Lib_get_fy_size();

    // 取込先画像オブジェクトの確保
    himg[0] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );
    himg[1] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );

    // 同時取込の設定
    Lib_set_sim_channel( SIM_CH0 | SIM_CH1 );

    // 取込実行
    Lib_fie_freeze( himg, 2 );
}
```

留意事項

○本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
○本ライブラリは取込の実行を行うだけの関数です。モードの切り替えには別途Lib_set_xvideo_input関数を使用ください。
○その他キャプチャボード・カメラ毎の留意事項に関しては、Lib_freezeの項を参照ください。

Lib_fie_xfreeze

機　能

FIE画像オブジェクトへの拡張画像取込

形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_xfreeze(FHANDLE *himg, INT num);
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

画像取込指示を発行し、カメラから画像を入力します。
関数は取込指示発行後に制御を戻します。取込の完了確認には Lib_get_video_status 関数を使用ください。

① ***himg** は取込先の FIE 画像オブジェクトを格納した配列の先頭アドレスです。

取り込まれた画像は、配列の先頭インデックスから若い番号順に格納されます。

		himg[]
・単チャンネル取込の場合	himg[0]	カレントチャンネルの画像
・1ch, 2ch 同時取込の場合	himg[0]	1ch の画像
	himg[1]	2ch の画像
・1ch, 3ch 同時取込の場合	himg[0]	1ch の画像
	himg[1]	3ch の画像

取込先のFIE画像オブジェクトは、下記の条件を満たしている必要があります。
- 画像タイプがF_IMG_UC8
- チャネル数が1
- 画像サイズが取込サイズと等しいこと

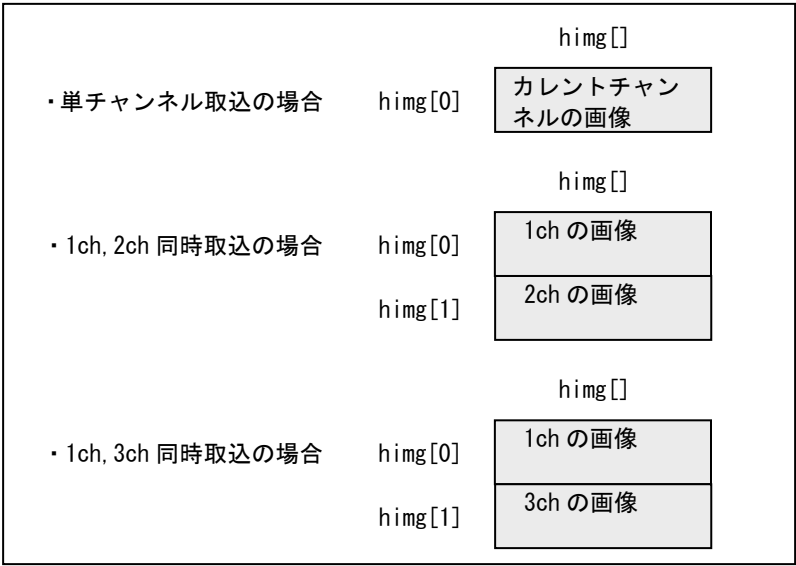
また、指定するFIE画像オブジェクトはルート画像でなくてはいけません。

② ***num**は***himg**で指定した取込先FIE画像オブジェクトの配列個数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。

例

```
#include "f_video.h"
#include "f_stdlib.h"
#include "f_fie.h"
#include "f_fieutil.h"

/* カレントチャネルの拡張取込実行 */
void freeze( void )
{
    FHANDLE himg = NULL;

    // 取込サイズ取得
    INT fx = Lib_get_fx_size();
    INT fy = Lib_get_fy_size();

    // 取込先画像オブジェクトの確保
    himg = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );

    // 取込実行
    Lib_fie_xfreeze( &himg, 1 );

    while( FINISHED_TRANSMIT != Lib_get_vide_status( Lib_get_video_channel() ) )
    {
          // 取込完了まで待機します。
    }
}

/* 2ch 同時拡張取込実行 */
void 2ch_freeze( void )
{
    FHANDLE himg[2] = { NULL, NULL };

    // 取込サイズ取得
    INT fx = Lib_get_fx_size();
    INT fy = Lib-get_fy_size();

    // 取込先画像オブジェクトの確保
    himg[0] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );
    himg[1] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );

    // 同時取込の設定
    Lib_set_sim_channel( SIM_CH0 | SIM_CH1 );
```

```
    // 取込実行
    Lib_fie_xfreeze( himg, 2 );

    while( FINISHED_TRANSMIT != Lib_get_vide_status( Lib_get_video_channel() ) )
    {
        // 取込完了まで待機します。
    }

}
```

留意事項

〇本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
〇本ライブラリは取込の実行を行うだけの関数です。モードの切り替えには別途Lib_set_xvideo_input 関数を使用ください。
〇その他キャプチャボード・カメラ毎の留意事項に関しては、Lib_xfreeze の項を参照ください。

Lib_fie_frame_chain_freeze

機　能

FIE画像オブジェクトへの連続取込

形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_frame_chain_freeze( FHANDLE *himg, INT num );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カレントビデオ入力チャネルの連続画像を、引数のFIE画像オブジェクト配列に順次入力します。

① ***himg*** は取込先のFIE画像オブジェクトを格納した配列の先頭アドレスです。

取込先のFIE画像オブジェクトは、下記の条件を満たしている必要があります。
- 画像タイプがF_IMG_UC8
- チャネル数が1
- 画像サイズが取込サイズと等しいこと

また、指定するFIE画像オブジェクトはルート画像でなくてはいけません。

② ***num***は***himg***で指定した取込先FIE画像オブジェクトの配列個数です。

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_video.h"
#include "f_stdlib.h"
#include "fie.h"
#include "f_fieutil.h"

#define START_MEMORY_NO     0
#define FRAME_SU            16

void frame_chain( void
{
        INT i;
FHANDLE himg[FRAM_SU];

    // 取込サイズ取得
INT fx = Lib_get_fx_size();
INT fy = Lib_get_fy_size();

    // 取込先画像オブジェクトの確保
    for( i = 0; i < FRAME_SU; i++ )
    {
        himg[i] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );
    }

    // 取込開始
    Lib_fie_frame_chain_freeze( himg, FRAME_SU );
}
```

留意事項

○本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
○本ライブラリは取込の実行を行うだけの関数です。モードの切り替えには別途Lib_set_xvideo_input 関数を使用ください。
○その他キャプチャボード・カメラ毎の留意事項に関しては、Lib_frame_chain_freezeの項を参照ください。

Lib_fie_capt_grab_start

機　能

FIE画像オブジェクトへの連続取込の開始

形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_capt_grab_start( FHANDLE *himg, INT num );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

カレントビデオ入力チャネルの連続画像を、引数のFIE画像オブジェクト配列に順次入力します。配列の最後まで取り込みが完了すると、配列の最初のFIE画像オブジェクトに戻って取込動作を継続します。

連続取込動作を停止する場合には、Lib_cat_grab_end関数をコールください。

① ***himg*** は取込先のFIE画像オブジェクトを格納した配列の先頭アドレスです。

取込先のFIE画像オブジェクトは、下記の条件を満たしている必要があります。
- 画像タイプがF_IMG_UC8
- チャネル数が1
- 画像サイズが取込サイズと等しいこと

また、指定するFIE画像オブジェクトはルート画像でなくてはいけません。

② ***num***は****himg***で指定した取込先FIE画像オブジェクトの配列個数です。

戻り値

処理結果

値	定　数	意　味
0	NORMAL_RETURN	正常終了しました。
-1	ERROR_RETURN	異常終了しました。

例

```
#include "f_video.h"
#include "f_stdlib.h"
#include "fie.h"
#include "f_fieutil.h"

#define START_MEMORY_NO     0
#define FRAME_SU            16

void continuous_grab( void )
{
        INT i, x, y, index;
FHANDLE himg[FRAM_SU];

    // 取込サイズ取得
INT fx = Lib_get_fx_size();
INT fy = Lib_get_fy_size();

INT channel = Lib_get_video_channel();

    // 取込先画像オブジェクトの確保
    for( i = 0; i < FRAME_SU; i++ )
    {
        himg[i] = fnFIE_img_root_alloc( F_IMG_UC8, 1, fx, fy );
    }

    // 連続取込開始
     if( NORML_RETURN == Lib_fie_capt_grab_start( himg, FRAME_SU ) )
     {
         while(1)
         {
             Lib_capt_wait_event( channel, 0 );
             Lib_cat_get_fram_index( channel, &index );
             index -= 1; if( index < 0 ) index = FRAME_SU - 1;

             // 何らかの処理
             if( 0 != Lib_see_current_position( &x, &y ) )
                break;
         }

         // 連続取込の終了
         Lib_capt_grab_end( channel );
     }
}
```

留意事項

〇本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
〇本ライブラリは取込の実行を行うだけの関数です。モードの切り替えには別途Lib_set_xvideo_input 関数を使用ください。
〇その他キャプチャボード・カメラ毎の留意事項に関しては、Lib_capt_grab_start の項を参照ください。

Lib_fie_transmit

機　能

カレントウィンドウへのFIE画像オブジェクトの描画

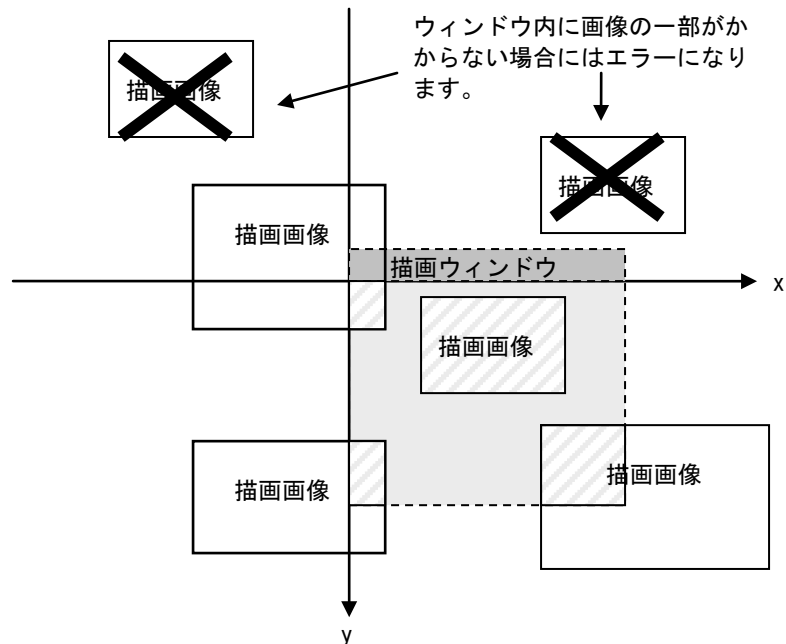
形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_transmit( FHANDLE himg, INT x, INT y );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

引数himgに入力されたFIE画像オブジェクトを、引数x,yを始点としてカレントウィンドウのGRAY_PLANEに描画します。描画するFIE画像オブジェクトは、ルート画像でもチャイルド画像でも構いません。

描画始点 x,y は、ウィンドウの左端を(0, 0)とした座標値で指定します。
負の値を入力した場合には、入力値分だけ x 値の場合には左に、y 値の場合には上に描画始点位置が移動します。ウィンドウに表示されるのは、ウィンドウ内に収められている領域だけになります。

① *himg* はウィンドウに表示するFIE画像オブジェクトです。
表示画像のタイプとチャンネルは下表の条件を満たしている必要があります。

画像タイプ	チャンネル数	備考
F_IMG_UC8	1	グレイ 8bit
F_IMG_UC8	3	RGBカラー
F_IMG_BIN	1	二値画像
F_IMG_RGBQUAD	1	RGBカラー

② *x* は描画先の x 座標始点です。
③ *y*は描画先のy座標始点です。

描画始点 x, y には、下記の制限があります。
・ x ＜ ウィンドウの横サイズ
・ y ＜ ウィンドウの縦サイズ

戻り値

処理結果

値	定　　数	意　　味
0	NORMAL_RETURN	正常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。
-10003	ERR_NO_MEMORY	メモリ不足

例

```
#include "fie.h"
#include "f_fieutil.h"

// bitmapファイルを読込み、表示を行います。
void bitmap_transmit( void )
{
        FHANDLE himg = NULL;

    // bitmapファイルの読み込み
    fnFIE_load_bmp( "/fs0/sample.bmp", &himg, F_COLOR_IMG_TYPE_RGBQ );

    // 画像の表示
    Lib_fie_transmit( himg, 0, 0 );
}


// bitmapファイルを読込み後、切り出し表示を行います。
void bitmap_transmit( void )
{
        FHANDLE himg = NULL;
FHANDLE hchild = NULL;

    // bitmapファイルの読み込み
    fnFIE_load_bmp( "/fs0/sample.bmp", &himg, F_COLOR_IMG_TYPE_RGBQ );

    // 切り出し画像オブジェクトの作成
    hchild = fnFIE_img_child_alloc( himg, 100, 100, 200, 200 );

    // 画像の表示
    Lib_fie_transmit( hchild, 0, 0 );
}
```

留意事項

〇本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
〇本ライブラリの描画は GRAY_PLANE 上に行われます。予め Lib_display_control 関数にて GRAY_PLANE を有効にしておいてください。

Lib_fie_strech_transmit

機　能

拡大縮小機能付きカレントウィンドウへのFIE画像オブジェクトの描画

形　式

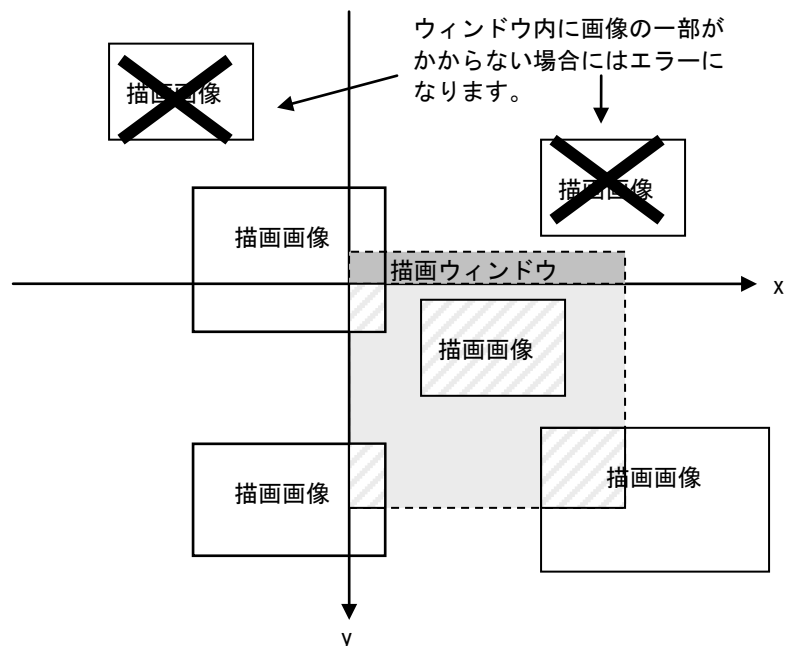
```
#include "f_fieutil.h"
int FVLCALL Lib_fie_strech_transmit( FHANDLE himg, INT x, INT y, INT width, INT height );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

引数himgに入力されたFIE画像オブジェクトを、引数x,yを始点としてカレントウィンドウのGRAY_PLANEに描画します。描画される画像は引数のwidth,heightのサイズに拡大、または縮小されて表示されます。描画するFIE画像オブジェクトは、ルート画像でもチャイルド画像でも構いません。

関数の内部で拡大・縮小用のバッファを確保するため、あまり大きいサイズを指定するとメモリ不足エラーになりますのでご注意ください。

描画始点 x,y は、ウィンドウの左端を(0, 0)とした座標値で指定します。
負の値を入力した場合には、入力値分だけ x 値の場合には左に、y 値の場合には上に描画始点位置が移動します。ウィンドウに表示されるのは、ウィンドウ内に収められている領域だけになります。

① ***himg*** はウィンドウに表示するFIE画像オブジェクトです。
表示画像のタイプとチャンネルは下表の条件を満たしている必要があります。

画像タイプ	チャンネル数	備考
F_IMG_UC8	1	グレイ 8bit
F_IMG_UC8	3	RGBカラー
F_IMG_BIN	1	二値画像
F_IMG_RGBQUAD	1	RGBカラー

② ***x***は描画先のx座標始点です。
③ ***y***は描画先のy座標始点です。

描画始点 x, y には、下記の制限があります。
・ x ＜ ウィンドウの横サイズ
・ y ＜ ウィンドウの縦サイズ

④ ***width***は描画の横サイズです。（ 1以上 ）
⑤ ***height***は描画の縦サイズです。（ 1以上 ）

戻り値

処理結果

値	定数	意味
0	NORMAL_RETURN	正常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。
-10003	ERR_NO_MEMORY	メモリ不足

例

```
#include ″fie.h″
#include ″f_fieutil.h″

// 4種のbitmapファイルを読込み、表示を行います。
void bitmap_transmit( void )
{
        FHANDLE himg[4] = { NULL, NULL, NUL, NULL};

// カレントウィンドウのサイズ取得
INT dx = Lib_get_dx_size();
INT dy = Lib_get_dy_size();

    // bitmapファイルの読み込み
    fnFIE_load_bmp( “/fs0/sample1.bmp”, &himg[0], F_COLOR_IMG_TYPE_RGBQ );
    fnFIE_load_bmp( “/fs0/sample2.bmp”, &himg[1], F_COLOR_IMG_TYPE_RGBQ );
    fnFIE_load_bmp( “/fs0/sample3.bmp”, &himg[2], F_COLOR_IMG_TYPE_RGBQ );
    fnFIE_load_bmp( “/fs0/sample4.bmp”, &himg[3], F_COLOR_IMG_TYPE_RGBQ );

    // 画像の表示 左上
    Lib_fie_transmit( himg[0], 0, 0, dx/2, dy/2 );

    // 画像の表示 右上
    Lib_fie_transmit( himg[1], dx/2-1, 0, dx/2, dy/2 );

    // 画像の表示 左下
    Lib_fie_transmit( himg[2], 0, dy/2-1, dx/2, dy/2 );

    // 画像の表示 右下
    Lib_fie_transmit( himg[3], dx/2-1, dy/2-1, dx/2, dy/2 );

}
```

留意事項

○本ライブラリを使用する前に、必ずfnFIE_setup関数をコールください。
○本ライブラリの描画はGRAY_PLANE上に行われます。予めLib_display_control関数にてGRAY_PLANEを有効にしておいてください。

Lib_fie_get_img_fhandle

機　能

メモリ番号のFIE画像オブジェクトへの変換

形　式

```
#include "f_fieutil.h"
FHANDLE  FVLCALL Lib_fie_get_img_fhandle( INT mem_no );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

引数のメモリ番号で指定されているメモリ領域を、FHANDLEでアクセスできるように変換します。
本関数使用後に確保したFIE画像オブジェクトは、fnFIE_free_object関数にて開放してください。

また、本関数はメモリの再確保を行うものではありません。
引数のmem_noは、予めLib_alloc_gray_memory、またはLib_alloc_bin_memory関数にてメモリ領域が確保されていなくてはいけません。また、確保されたメモリ領域は、別途Lib_free_gray_memory/Lib_free_bin_memory関数にて開放してください。

① ***mem_no*** はFIE画像オブジェクトに変更するメモリ番号です。
CURRENT_MEMORYを指定した場合には、グレイメモリのFHANDLEが返されます。

戻り値

処理結果

値	意味	
NULL	異常終了	
NULL以外	正常終了	確保済みメモリのFIE画像オブジェクトのハンドル

例

メモリを確保し、FIEライブラリで処理を行います。

```
#include "fie.h"
#include "f_fieutil.h"

void do_Liblaplacian_with_fieimage( void )
{
FHANDLE hsrc = NULL;
FHANDLE hdst = NULL;
INT src_mem_no, dst_mem_no;

// メモリの確保
src_mem_no = Lib_alloc_gray_memory();
dst_mem_no = Lib_alloc_gray_memory();

    // 確保したメモリをFIEの画像オブジェクトに変換
hsrc = Lib_fie_get_img_fhandle( src_mem_no );
hdst = Lib_fie_get_img_fhandle( dst_mem_no );

// 処理対象のFIE画像オブジェクトにbitmapファイルを読み込む( グレイ画像 )
// ※読み込む画像ファイルは必ず確保したメモリと同じサイズにしてください。
```

```
fnFIE_load_bmp( “/fs0/gray.bmp”, &hsrc, F_COLOR_IMG_TYPE_UC8 );

// laplacian実行
    // このとき、hdstのメモリへの変更内容は直接、先に確保されたdst_mem_noへ反映さ
    れます。
fnFIE_laplacian( hsrc, hdst, 1, F_BORDER_NONE, 0 );

// 結果画像の表示
Lib_xvideo_transmit( dst_mem_no, GRAY_PlANE );

/*
  あるいは、メモリの内容が同じであるため、下記方法でも同じ画像がウィンドウに
     表示されます。
      Lib_fie_transmit( hdst, 0, 0 );
*/

// 確保したメモリ番号の開放
    Lib_free_gray_memory( dst_mem_no );
Lib_free_gray_memory( src_mem_no );

// 確保したFIE画像オブジェクトの開放
fnFIE_free_object( hsrc );
fnFIE_free_object( hdst );
}
```

留意事項

ありません。

Lib_fie_attach_image_memory

機　能

FIE画像オブジェクトのメモリ番号への変換

形　式

```
#include "f_fieutil.h"
int FVLCALL Lib_fie_attach_image_memory( FHANDLE himg, INT ch );
```

901	902	903	904	FVL/LNX	高分解能
				○	

解　説

引数のFIE画像オブジェクト内に確保されているメモリ領域を、メモリ番号でアクセスできるように変換します。
本関数使用後に確保したメモリ番号は、Lib_free_gray_memory関数等で開放処理を行ってください。

本関数は、メモリの再確保を行うものではありません。
引数のhimgは、予めfnFIE_img_root_alloc関数にてFIE画像オブジェクトのメモリ領域が確保されていなくてはいけません。また、確保されたhimgは、別途fnFIE_free_object関数で開放してください。

② ***himg*** はメモリ番号に変更するFIE画像オブジェクトです。

指定するFIE画像オブジェクトは、下記の条件を満たしている必要があります。
- 画像タイプがF_IMG_UC8, F_IMG_BIN
- 画像サイズが32の倍数であること
    - 画像のstepとwidthが等しいこと

また、指定するFIE画像オブジェクトはルート画像でなくてはいけません。

注意：
FIE画像オブジェクトの画像タイプがF_IMG_BINの場合には、パディングサイズによって画像のstepとwidthがイコールにならないことがあります。
そのような場合には、fnFIE_img_root_alloc関数でFIE画像オブジェクトを確保するまえに、fnFIE_img_set_padding_size関数でパディングサイズを調整してください。

② ***ch***は、***himg***のチャンネル番号です。

戻り値

処理結果

値	定　　数	意　　味
0 <		変換されたメモリ番号
-1	ERROR_RETURN	異常終了しました。
-10002	ERR_ILLEGAL_PARAM	引数に異常があります。
-10100	ERR_ILLEGAL_IMAGE_TYPE	対応していない画像タイプのFIE画像オブジェクトが入力されました。
-10101	ERR_ILLEGAL_IMAGE_SIZE	対応していない画像サイズのFIE画像オブジェクトが入力されました。(widthが32の倍数でない, stepとwidthが同じでない)

例

```
#include ″f_video.h″

#include ″fie.h″
#include ″f_fieutil.h″

#define START_MEMORY_NO      0
#define FRAME_SU             16

// FIE画像オブジェクトを確保し、Lib_laplacianで処理を行います。
void do_Liblaplacian_with_fieimage( void )
{
FHANDLE hsrc = NULL;
FHANDLE hdst = NULL;

INT src_mem_no, dst_mem_no;

    // FIE画像オブジェクトの確保
hsrc = fnFIE_img_root_alloc( F_IMG_UC8, 1, 512, 480 );
hdst = fnFIE_img_root_alloc( F_IMG_UC8, 1, 512, 480 );

// 処理対象のFIE画像オブジェクトにbitmapファイルを読み込む( グレイ画像 )
fnFIE_load_bmp( “/fs0/gray.bmp”, &hsrc, F_COLOR_IMG_TYPE_UC8 );

// FIE画像オブジェクトをメモリ番号に変換
    // このとき、対象のメモリ番号のメモリの中身は、先に確保したFIE画像オブジェク
    // トのものと同じになります。
src_mem_no = Lib_fie_attach_image_memory( hsrc, 0 );
dst_mem_no = Lib_fie_attach_image_memory( hdst, 0 );

// laplacian実行
Lib_laplacian( src_mem_no, dst_mem_no );

// 結果画像の表示
Lib_xvideo_transmit( dst_mem_no, GRAY_PlANE );

/*
   あるいは、メモリの内容が同じであるため、下記方法でも同じ画像がウィンドウに
      表示されます。
        Lib_fie_transmit( hdst, 0, 0 );
*/

// 確保したメモリ番号の開放
    Lib_free_gray_memory( dst_mem_no );
Lib_free_gray_memory( src_mem_no );

// 確保したFIE画像オブジェクトの開放
fnFIE_free_object( hsrc );
fnFIE_free_object( hdst );
}
```

留意事項

ありません。

3. Lib_capt_set_reg　補足説明

FVL/LNX で FVC06,FHC3312,FHC3313 を使用した際に、Lib_capt_set_reg 関数で使用できるパラメータについて説明をします。

本書に記載されているパラメータの使用方法につきましては、「90X 基本ライブラリ説明書～画像入力編～」の Lib_capt_set_reg 関数、及び、Lib_capt_ｇet_reg 関数の項を参照ください。

3.1　FVC06

3.1.1 取り込みに関するパラメータ

FVC06 の取り込みに関するパラメータを設定します。

FVC06_VIDEO_TIMEOUT
取り込みタイムアウト時間を設定します。単位は ms です。
設定範囲：0～4294967295

FVC06_VIDEO_HORIZONTALSIZE
水平入力サイズを設定します。単位は pixel です。カメラの入力サイズに合わせて指定して下さい。ただし、この値は画像メモリのサイズではなく、1TAP あたりの入力サイズとなります。そのため、次のように計算してください。
また、画像処理ライブラリの制約上、画像メモリ横サイズが 32 画素の倍数になるようにしなければいけません。

カメラの種類	入力値
1TAP のカメラ	有効画素数の横サイズをそのまま。（32 の倍数）
2TAP のカメラ	有効画素数の横サイズ÷2（16 の倍数）
3TAP のカラーカメラ	有効画素数の横サイズをそのまま。（32 の倍数）
4TAP のカメラ	有効画素数の横サイズ÷4（8 の倍数）

設定範囲：0～16385

FVC06_VIDEO_VERTICALSIZE
垂直入力サイズを指定します。単位は pixel です。
カメラの入力サイズに合わせて指定して下さい。カメラの入力 TAP による計算はありません。
設定範囲：0～65535

FVC06_VIDEO_HORIZONTALOFFSET
水平方向オフセットを設定します。カメラの仕様書（タイミングチャート）を参考に指定して下さい。設定値は LineValid(LVAL)の立ち上がりから、映像出力までのカウントを指定して下さい。このとき、映像出力期間の中心を入力するように指定してください。この値はノーマルモード時に使用されます。
設定範囲：0～65535

FVC06_VIDEO_VERTICALOFFSET
垂直方向オフセットを設定します。カメラの仕様書（タイミングチャート）を参考に指定してください。設定値は FrameValid(FVAL)の立ち上がりから、映像出力までのカウントを指定します。この値はノーマルモード時に使用されます。
設定範囲：0～65535

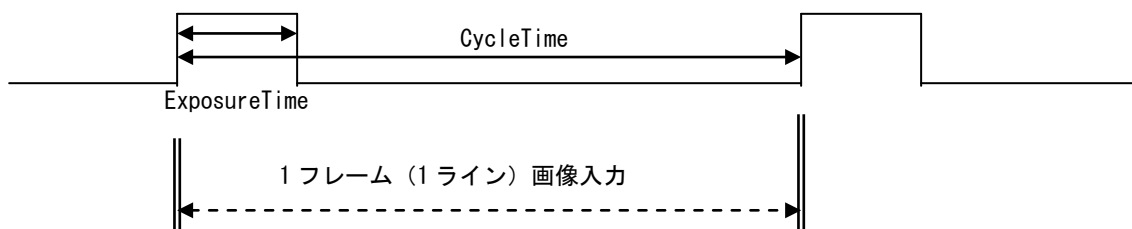
FVC06_VIDEO_VERTICALOFFSETMD
連続取り込みの場合に、垂直オフセットの設定を各フレーム全てに対して行うか、最初のフレームのみに行うかを設定します。
0 を指定すると、全フレームに対して垂直オフセットが有効になります。1 を指定すると、最初のフレームに対しては垂直オフセットが有効になりますが、連続取り込みの 2 フレーム目以降は、垂直オフセットが 0 となります。
エリアカメラの場合、常に 0 を指定してください。ラインセンサの場合、連続取り込みを行うと一般的には最初のフレームのみ垂直オフセットを行い、2 フレーム目以降は前フレームからライン抜けなく入力することが多いので垂直オフセットを 0 にしたい場合が多くなります。しかし、エンコーダの Z 相スタートや、外部トリガスタートでは全フレームに垂直オフセットが効いたほうが良い場合があります。

FVC06_VIDEO_ENCODERENABLE
エンコーダを使用するかどうかを指定します。本パラメータに 1 を指定すると、エンコーダの使用が有効になります。0 を指定すると、エンコーダの使用が無効になります。

3.1.2　トリガに関するパラメータ

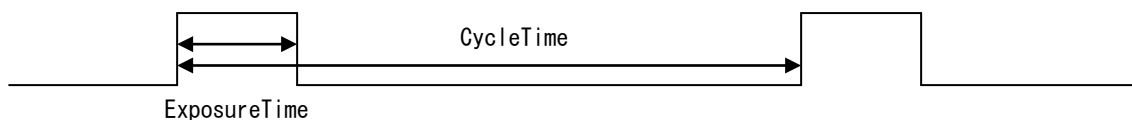
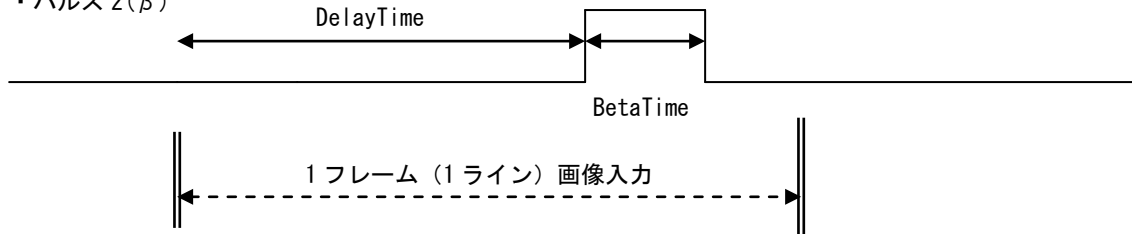
FVC06のトリガを使用した取込に関するパラメータを設定します。

FVC06でトリガを使用した取込を行う場合、カメラの取り込み方法によって下図（図1、図2）のように取り込みパルスを制御して取り込みを行います。
本項で説明するパラメータでは、図中のExposureTimeや、CycleTimeといった値の変更を行います。

1. 1パルストリガ取り込みのカメラの場合

【図１：１パルストリガ取り込みのカメラの場合】

2. カメラが2パルストリガ取り込みの場合

・パルス1(α)

・パルス2(β)

【図2：2パルストリガ取り込みのカメラの場合】

・ExposureTime　：FVC06_TRIGGER_EXPOSURETIMEパラメータで設定される値です。
・CycleTime　　：FVC06_TRIGGER_CYCLETIMEパラメータで設定される値
・BetaTime　　　：FVC06_TRIGGER_BETATIMEパラメータで設定される値
・DelayTime　　：FVC06_TRIGGER_DELAYTIMEパラメータで設定される値

FVC06_TRIGGER_TRIGGER
ノーマルモードか、トリガモードなのかを指定します。
0を指定するとノーマルモードで、1を指定するとトリガモードになります。

FVC06_TRIGGER_EXTERNALTRIGGER
外部トリガを有効にする場合には1を、無効にする場合には1を設定します。

FVC06_TRIGGER_EXPOSURETIME
CCパルス出力期間ExposureTime（図1、図2参照）を設定します。単位はusです。
一般的には露光時間の設定と同じ意味になります。
設定範囲：0～4294967295

FVC06_TRIGGER_BETATIME
CCパルス出力期間BetaTime（図2参照）を設定します。単位はusです。
2パルストリガ取り込みカメラの使用時に設定します。通常、2パルストリガ取り込みカメラの場合には、ExposureTimeとBetaTimeに短いパルス（カメラ仕様書に記載されている時間）を出力するようにします。
設定範囲：0～4294967295

FVC06_TRIGGER_DELAYTIME
CCパルス出力間隔DelayTime（図2参照）を設定します。単位はusです。
ExposureTimeとBetaTimeの間隔を指定します。BetaTimeが使用されるときにのみ使われます。
2パルストリガ取り込みカメラの場合には露光時間として使用されることがあります。
設定範囲：0～4294967295

FVC06_TRIGGER_SIMTRIGGER
同時トリガモードの設定をします。1を指定するとトリガをチャンネル0とチャンネル1の両方から出力します。同時取り込みを使用する場合は1を指定して下さい。この設定はCH0側のチャネルからしか設定できません。

FVC06_TRIGGER_EXTRIGGEROR
同時トリガモード時のトリガの論理和設定をします。本パラメータを1にすると、チャンネル0、チャンネル1のどちらに外部トリガ入力してもトリガと見なされ、画像入力を開始することが可能です。この設定はチャンネル0側からしか設定できません。

FVC06_TRIGGER_CYCLETIME
トリガサイクル（CycleTime（図1、図2参照））を設定します。単位はusです。

(例) CV-M9CL Partial=NONE, Binning=NONEの場合

CV-M9CLは、パルス幅トリガモードの場合、非同期蓄積で行います。
「非同期蓄積での最小のトリガ期間は（露光時間+1FVAL+3LVAL）以上」と記載されております。
1LVAL = 42.07us
1FVAL = 788LVAL
であるため、
　　1FVAL+3LVAL = 791LVAL = 33277.37us
となります。
よって、トリガ出力期間+33277.37us が最小のトリガ周期であり、パルス幅モードでのトリガを行う場合、本設定は33278（小数点以下が存在するときは必ず切り上げ）を指定します。

エリアカメラで取り込みをトリガモードに指定した場合には、連続取り込みの取り込み間隔を本パラメータによって指定します。
ラインセンサカメラ使用時には1ラインのスキャンレート値になります。

設定範囲：0～4294967295

FVC06_TRIGGER_MASKLINENUM
トリガの無効期間を設定します。単位はusです。
本パラメータで設定した期間中はトリガを受けつけません。
設定範囲：0～4294967295

FVC06_TRIGGER_PGMODE
トリガのリトライモードを設定します。
本パラメータが1に設定されていた場合には、トリガをFVC06_TRIGGER_CYCLETIMEモードで設定した間隔で連続的に出力します。

3.1.3 カメラコントロールに関するパラメータ

FVC06 のカメラコントロール(CC)に関するパラメータを設定します。
このパラメータは、各チャネルの CC の設定を行います。通常、LNX ではシステム起動時に指定のカメラに対して CC のパラメータを適切に設定するため、本パラメータ群を変更する必要性はありません。

FVC06_CC_ENABLE01
FVC06_CC_ENABLE02
FVC06_CC_ENABLE03
FVC06_CC_ENABLE04
CC1～CC4 の Enable 設定をします。
1 を指定すると、トリガモードで画像入力時にトリガ出力が行われます。

FVC06_CC_MODE01
FVC06_CC_MODE02
FVC06_CC_MODE03
FVC06_CC_MODE04
CC のトリガ接続モードを指定します。
FVC06 はトリガ発生器を２つ持っています。トリガ発生器は「トリガに関するパラメータ」で説明されているようにαとβの２種類のトリガを発生できます。また、トリガ発生器からのトリガとは別に 2 つの外部トリガを直接出力することができます。このパラメータは CC1~CC4 にたいして、どのトリガを使用するかを設定します。

入力値：

値	意　　　味
0	内部トリガ0-αを使用
1	内部トリガ0-βを使用
2	内部トリガ1-αを使用
3	内部トリガ1-βを使用
4	外部トリガ入力0を使用
5	外部トリガ入力1を使用

FVC06_CC_POSITIVE01
FVC06_CC_POSITIVE02
FVC06_CC_POSITIVE03
FVC06_CC_POSITIVE04
CC 信号の立ち上がり、立ち下がりのどちらを検出するかを指定します。0 を指定すると立ち下がりとなり、1 を指定すると立ち上がりとなります。

FVC06_CC_ENABLEEXP
外部露光出力の Enable 設定をします。

FVC06_CC_MODEEXP
外部露光出力の CC のトリガ接続モードを指定します。

FVC06_CC_POSITIVEEXP
外部露光出力の立ち上がり、立ち下がりを設定します。

3.1.4　エンコーダに関するパラメータ

エンコーダに関する設定です。FVC06_VIDEO_ENCODERENABLE パラメータに 1 を設定した場合に限り、以降のパラメータが有効となります。

FVC06_ENCODER_ENCSIGNAL
エンコーダの A 相, AB 相の使用を指定します。

入力値：

値	意　　味
0	AB 相を使用
1	A 相を使用

FVC06_ENCODER_ENCPULSE
エンコーダの回転方向を指定します。FVC06_ENCODER_ENCSIGNAL パラメータに 0 を指定したとき(AB 相)のときのみ有効です。

入力値：

値	意　　味
0	CW
1	CCW

FVC06_ENCODER_ENCMLT
エンコーダサンプリングレートを指定します。
FVC06_ENCODER_ENCPULSE パラメータに 0(AB 相)を指定していた場合には、0～2 が設定可能です。FVC06_ENCODER_ENCPULSE パラメータに 1 (A 相)を指定していた場合には、0～1 が設定可能です。

入力値：

値	意　　味
0	1 倍
1	2 倍
2	4 倍（AB 相指定時のみ有効）

FVC06_ENCODER_ENCSTART
エンコーダカウントのスタートタイミングを設定します。

入力値：

値	意　　味
0	CPU ・画像入力の指示を受け次第、エンコーダのカウントを始めます。
1	外部トリガ ・画像入力の指示を受けると外部トリガ入力を待ちます。外部トリガの入力を検出すると、エンコーダのカウントを始めます。
2	Z 相 ・画像入力の指示を受けると、Z 相入力を待ちます。Z 相の入力を検出すると、エンコーダのカウントを始めます。

FVC06_ENCODER_COMPREG1
エンコーダカウントの遅延パルス数を設定します。0 以上の値を入力してください。
エンコーダカウントスタートから、本パラメータで設定されたカウント数分待ってから画像入力を開始します。
設定範囲：0～4294967295

FVC06_ENCODER_COMPREG2
エンコーダのカウント周期を設定します。1 以上の値を入力してください。
エンコーダのパルスカウント値が本パラメータで設定された値と一致した場合に、画像の入力を行います。
設定範囲：0～65535

FVC06_ENCODER_ENCINIT
取込み完了時のエンコーダ動作を設定します。
0 を指定するとエンコーダ停止。1 を設定するとエンコーダ動作を継続します。

3.1.5　FVC06 の ini ファイル仕様

FVC06 で使用している ini ファイルの各キーの値と、Lib_capt_set_reg 関数にて設定できる値との対応表を以下に示します。

Lib_capt_set_reg での対応値がないものは、ini ファイルによってカメラ特有に設定されている値です。
そのため、誤って ini ファイルの編集を行うと取込ができなくなる可能性がありますのでご注意ください。

ini ファイルの項目	Lib_capt_set_reg での対応値	備考
[Version]		
Version	−	
[Camera]		
CameraName	−	
CameraMaker	−	
[Device]		
Kind	−	
ID	−	
Configuration	−	
CH	−	
[Video]		
Timeout	FVC06_VIDEO_TIMEOUT	タイムアウト値
HorizontalSize	FVC06_VIDEO_HORIZONTALSIZE	X サイズ
VerticalSize	FVC06_VIDEO_VERTICALSIZE	Y サイズ
HorizontalOffset	FVC06_VIDEO_HORIZONTALOFFSET	X オフセット
VerticalOffset	FVC06_VIDEO_VERTICALOFFSET	Y オフセット
HorizontalOffsetTrg	−	
VerticalOffsetTrg	−	
VerticalOffsetMD	FVC06_VIDEO_VERTICALOFFSETMD	Y オフセット値の連続フラグ
LateDMA	−	
BitWidth	−	
Force8	−	
ReverseOrder	−	
TapCount	−	
LineSensorMode	−	
EncoderEnable	FVC06_VIDEO_ENCODERENABLE	エンコーダの有効・無効
ScanMode	−	
DVALEnable	−	
PackingMode	−	
Alignment16	−	
FormatType	−	
[Trigger]		
Trigger	FVC06_TRIGGER_TRIGGER	トリガの有効・無効
ExternalTrigger	FVC06_TRIGGER_EXTERNALTRIGGER	外部トリガの有効・無効
ExposureTime	FVC06_TRIGGER_EXPOSURETIME	パルス 1 の露光時間
BetaTime	FVC06_TRIGGER_BETATIME	パルス 2 の露光時間
DelayTime	FVC06_TRIGGER_DELAYTIME	パルス 1 とパルス 2 の間隔
SimTrigger	FVC06_TRIGGER_SIMTRIGGER	同時トリガモード設定
ExTriggerOr	FVC06_TRIGGER_EXTIGGEROR	同時トリガモードの論理和設定
CycleTime	FVC06_TRIGGER_CYCLETIME	パルス 1 の出力間隔
PGMode	FVC06_TRIGGER_PGMODE	トリガのリトライモード
MaskLineNum	FVC06_TRIGGER_MASKLINENUM	トリガの無効期間設定
[CC]		
Enable01	FVC06_CC_ENABLE01	

Mode01	FVC06_CC_MODE01	
Positive01	FVC06_CC_POSITIVE01	
Enable02	FVC06_CC_ENABLE02	
Mode02	FVC06_CC_MODE02	
Positive02	FVC06_CC_POSITIVE02	
Enable03	FVC06_CC_ENABLE03	
Mode03	FVC06_CC_MODE03	
Positive03	FVC06_CC_POSITIVE03	
Enable04	FVC06_CC_ENABLE04	
Mode04	FVC06_CC_MODE04	
Positive04	FVC06_CC_POSITIVE04	
EnableEXP	FVC06_CC_ENABLEEXP	
ModeEXP	FVC06_CC_MODEEXP	
PositiveEXP	FVC06_CC_POSITIVEEXP	
[Encoder]		
EncSignal	FVC06_ENCODER_ENCSIGNAL	エンコーダの位相設定
EncPulse	FVC06_ENCODER_ENCPULSE	エンコーダの回転方向
EncMlt	FVC06_ENCODER_ENCMLT	エンコーダのサンプリングレート
EncStart	FVC06_ENCODER_ENCSTART	エンコーダの開始方法
CompReg1	FVC06_ENCODER_COMPREG1	比較レジスタ 1
CompReg2	FVC06_ENCODER_COMPREG2	比較レジスタ 2
EncInit	FVC06_ENCODER_ENCINIT	エンコーダの終了処理
[SIO]		
RxMode	–	
TxMode	–	
Baudrate	–	
RxBuffSize	–	
TxBuffsize	–	
EventRevData	–	
STX	–	カメラコマンドの STX
ETX	–	カメラコマンドの ETX
Option	–	カメラコマンドのオプション
[CmdInitial]		
cmd	–	3.3 項を参照ください
[CmdNormal]		
cmd	–	3.3 項を参照ください
[CmdTrigger]		
cmd	–	3.3 項を参照ください

3.2　FHC3312,FHC3313 に関するパラメータ

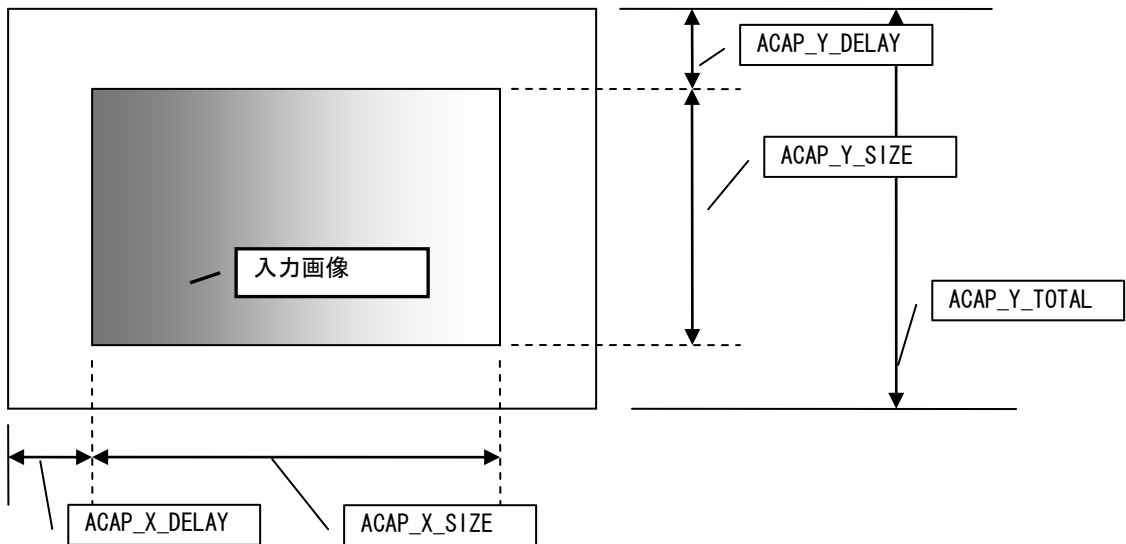
3.2.1　取り込みに関するパラメータ

ACAP_X_SIZE
水平方向の画像入力サイズです。32 の倍数である必要があります。

ACAP_Y_SIZE
垂直方向の画像入力サイズです。

ACAP_X_DELAY
水平方向の画像入力オフセット値です。

ACAP_Y_DELAY
垂直方向の画像入力オフセット値です。

ACAP_Y_TOTAL
垂直方向の画像入力トータルサイズです。

ACAP_TIME_OUT
取り込みタイムアウト時間を設定します。単位は ms です。

3.2.2　トリガに関するパラメータ

ACAP_EXP_EN
露光信号出力の有効・無効を設定します。

入力値：

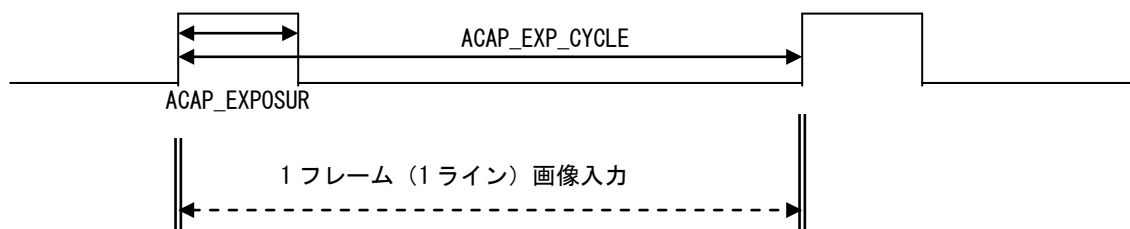
値	意　　　　味
0	無効
1	有効

ACAP_EXP_POL
露光信号の出力論理です。

入力値：

値	意　　　　味
0	正論理
1	負論理

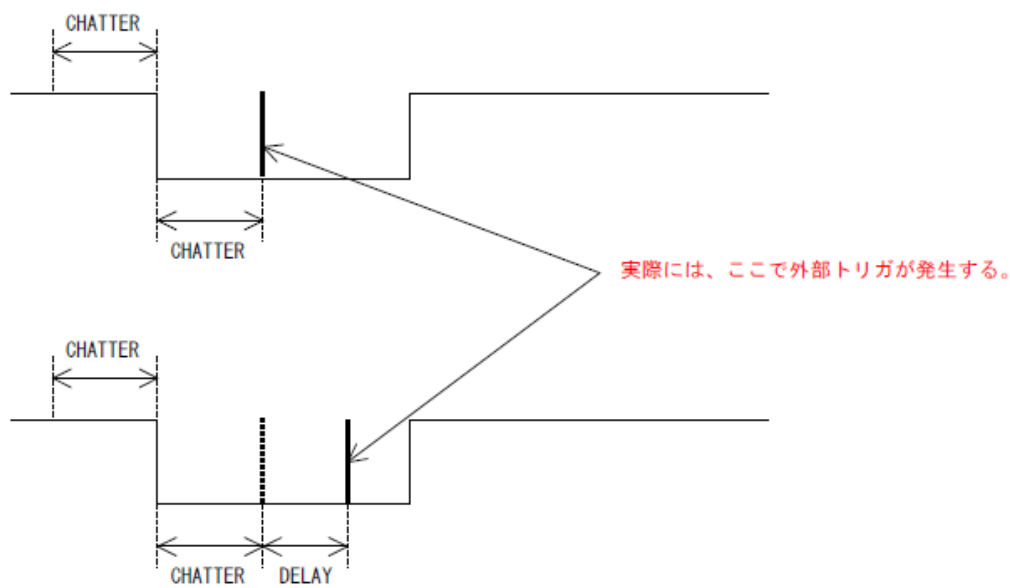
ACAP_EXP_CYCLE
露光制御信号の露光周期です。単位は us です。

ACAP_EXPOSURE
露光制御信号の露光時間です。単位は us です。

ACAP_EXT_EN
外部トリガの有効・無効設定です。

入力値：

値	意　　　　味
0	無効
1	有効[TTL]
2	有効[RS-422]

ACAP_EXT_CHATTER
外部トリガの無効期間を設定します。単位は us です。
下図を参照ください。

ACAP_EXT_DELAY
外部トリガの出力遅延時間を設定します。単位は us です。
下図を参照ください。

ACAP_STROBE_EN
ストロボの有効・無効を設定します。

入力値：

値	意　　味
0	無効
1	有効

ACAP_STROBE_DELAY
ストロボの発生遅延時間を設定します。単位は us です。
Lib_capt_set_strobe の項を併せて参照ください。

ACAP_STROBE_TIME
ストロボの発生時間を設定します。単位は us です。
Lib_capt_set_strobe の項を併せて参照ください。

3.2.3 カメラコントロールに関するパラメータ

ACAP_EXP_CC_OUT
露光信号の出力 CC を設定します。

入力値：

値	意　　　味
1	CC1
2	CC2
3	CC3
4	CC4

ACAP_CC1_LEVEL
ACAP_CC2_LEVEL
ACAP_CC3_LEVEL
ACAP_CC4_LEVEL
CC 出力の位相設定です。

入力値：

値	意　　　味
0	Positive(High 状態)
1	Negative(Low 状態)

3.2.4 エンコーダに関するパラメータ

ACAP_ENC_EN
エンコーダの有効・無効設定です。

入力値：

値	意　　　味
0	無効
1	有効

ACAP_ENC_START
エンコーダ用外部トリガの起動方法です。

入力値：

値	意　　　味
0	CPU
1	起動信号
2	一致信号

ACAP_ENC_MODE
エンコーダ用外部トリガの起動方法です。

入力値：

値	意　　　味
0	エンコーダスキャンモード
1	エンコーダライン選択モード

ACAP_ENC_PHASE
エンコーダの位相設定です。

入力値：

値	意　　　味
0	A 相
1	AB 相

ACAP_ENC_DIRECTION
エンコーダの回転方向の設定です。

入力値：

値	意　　　味
0	CW(時計回り)
1	CCW(反時計回り)

ACAP_ENC_ZPHASE_EN
Z 相の有効・無効を設定します。

入力値：

値	意　　　味
0	無効
1	有効

ACAP_ENC_COMPARE_1
比較レジスタ 1 を設定します。
エンコーダカウントの遅延パルス数を設定します。0 以上の値を入力してください。
エンコーダカウントスタートから、本パラメータで設定されたカウント数分待ってから画像入力を開始します。

ACAP_ENC_COMPARE_2
比較レジスタ 2 を設定します。
エンコーダのカウント周期を設定します。1 以上の値を入力してください。
エンコーダのパルスカウント値が本パラメータで設定された値と一致した場合に、画像の入力を行います。

3.2.5　FHC3312,FHC3313 の ini ファイル仕様

FHC3312,FHC3313 で使用している ini ファイルの各キーの値と、Lib_capt_set_reg 関数にて設定できる値との対応表を以下に示します。

Lib_capt_set_reg での対応値がないものは、ini ファイルによってカメラ特有に設定されている値です。
そのため、誤って ini ファイルの編集を行うと取込ができなくなる可能性がありますのでご注意ください。

ini ファイルの項目	Lib_capt_set_reg での対応値	備考
[Size]		
USER_X_SIZE	ACAP_X_SIZE	X サイズ
USER_Y_SIZE	ACAP_Y_SIZE	Y サイズ
USER_Y_TOTAL_SIZE	ACAP_Y_TOTAL	Ytoal サイズ
USER_X_FRONT_PORCH	ACAP_X_DELAY	X ディレイ
USER_Y_FRONT_PORCH	ACAP_Y_DELAY	Y ディレイ
[Bit]		
USER_GRABBER_BIT_DEPTH	-	
USER_CAMERA_BIT_DEPTH	-	
USER_PIXEL_DATA_SHIFT	-	
[CC]		
USER_CL_CC1_POLARITY	ACAP_CC1_LEVEL	CC1 レベル(位相)
USER_CL_CC2_POLARITY	ACAP_CC2_LEVEL	CC2 レベル(位相)
USER_CL_CC3_POLARITY	ACAP_CC3_LEVEL	CC3 レベル(位相)
USER_CL_CC4_POLARITY	ACAP_CC4_LEVEL	CC4 レベル(位相)
USER_CL_CC_OUT_NO	ACAP_EXP_CC_OUT	露光信号出力 CC
[CC Trigger]		
USER_SW_TRIGGER_ENABLE	ACAP_EXP_EN	露光信号出力　有効・無効
USER_SW_TRIGGER_POLARITY	ACAP_EXP_POL	露光信号　出力論理
USER_SW_TRIGGER_CYCLE	ACAP_EXP_CYCLE	露光周期 ini ファイルは 100ns 単位
USER_SW_TRIGGER_WIDTH	ACAP_EXPOSURE	露光時間 ini ファイルは 100ns 単位
[External Trigger]		
USER_EXTERNAL_TRIGGER_ENABLE	ACAP_EXT_EN	外部トリガ　有効・無効
USER_EXTERNAL_TRIGGER_POLAITY	-	
USER_EXTERNAL_TRIGGER_DELAY	ACAP_EXT_DELAY	外部トリガ　遅延時間
USER_EXTERNAL_TRIGGER_CHATTER	ACAP_EXT_CHATTER	外部トリガ　無効時間
[Encoder]		
USER_ENCODER_ENABLE	ACAP_ENC_EN	エンコーダ　有効・無効
USER_ENCODER_MODE	ACAP_ENC_MODE	エンコーダモード選択
USER_ENCODER_PHASE	ACAP_ENC_PHASE	エンコーダの位相
USER_ENCODER_PULSE	ACAP_ENC_DIRECTION	エンコーダの回転方向
USER_ENCODER_COMPARE_REG_1	ACAP_ENC_COMPARE_1	比較レジスタ 1
USER_ENCODER_COMPARE_REG_2	ACAP_ENC_COMPARE_2	比較レジスタ 2
USER_ENCODER_Z_PHASE	ACAP_ENC_ZPHASE_EN	Z 相使用選択
USER_ENCODER_START	ACAP_ENC_START	エンコーダ用外部トリガ起動選択
[Strobe]		
USER_STROBE_ENABLE	ACAP_STROBE_EN	ストロボ　有効・無効
USER_STROBE_TIME_COUNT	ACAP_STROBE_TIME	ストロボ出力幅
USER_STROBE_DELAY_COUNT	ACAP_STROBE_DELAY	ストロボ遅延出力
USER_STROBE_POLALITY	-	
[CameraLink]		
USER_X_TAPS_PER_CH	-	

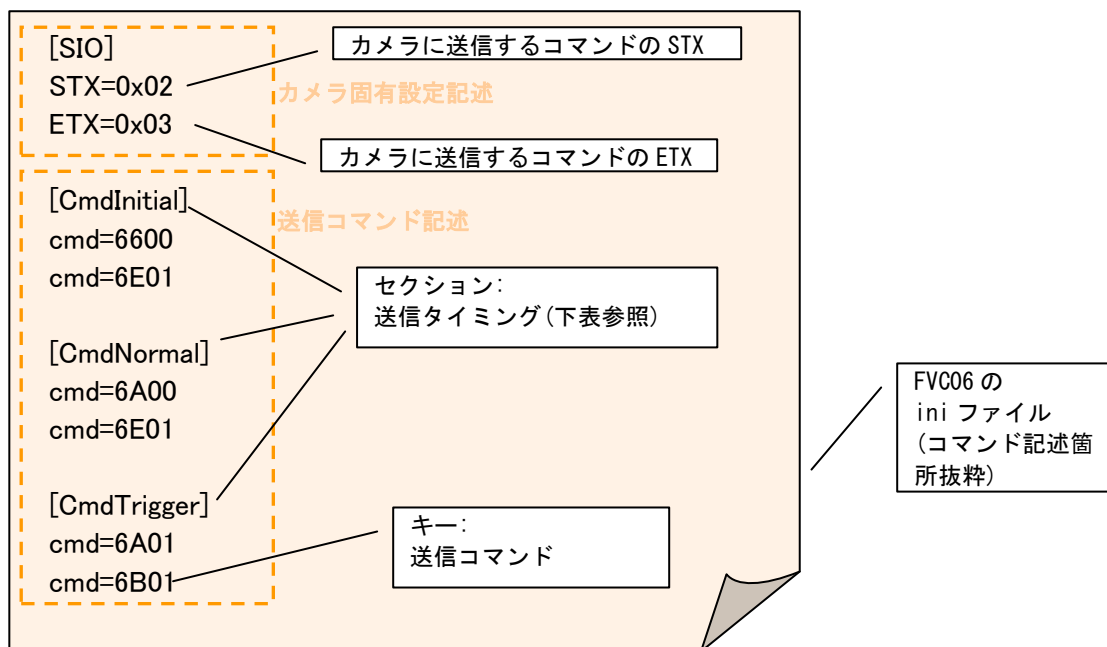
USER_CAMERALINK_TAP_DIRECTION	–	
USER_CAMERALINK_REARRANGEMENT_ENABLE	–	
USER_CAMERALINK_REARRANGEMENT_XSIZE	–	
USER_CAMERALINK_LVAL_DELAY	–	
USER_CAMERALINK_REARRANGEMENT_VALUE	–	
USER_CAMERALINK_TAP_DIRECTION_1	–	
USER_CAMERALINK_TAP_DIRECTION_2	–	
USER_CAMERALINK_TAP_DIRECTION_3	–	
USER_CAMERALINK_TAP_DIRECTION_4	–	
USER_CAMERALINK_TAP_DIRECTION_5	–	
USER_CAMERALINK_TAP_DIRECTION_6	–	
USER_CAMERALINK_TAP_DIRECTION_7	–	
USER_CAMERALINK_TAP_DIRECTION_8	–	
USER_CAMERALINK_LINE_REVERSE	–	
USER_HIGHCLIP_ENABLE	–	
[Bayer]		
USER_BAYER_ENABLE	–	
USER_BAYER_LUT_RED	–	
USER_BAYER_LUT_GREEN	–	
USER_BAYER_LUT_BLUE	–	
[Other]		
USER_INTERLACE_TYPE	–	
USER_MEMORY_NUM	–	
USER_TIMEOUT1	ACAP_TIME_OUT	タイムアウト
USER_REVERSE_DMA_ENABLE	–	
USER_DVAL_SIGNAL_ENABLE	–	
USER_ROLLING_SHUTTER_TRIGGER_ENABLE	–	
USER_2CAMERA_SYNC_MODE	–	
USER_DATA_INTERRUT_LINE	–	

3.3　ini ファイルのコマンド送信の仕様

LNX で使用している FVC06 や FHC シリーズのカメラリンク対応ボードでは、
- カメラの初期化時
- カメラのビデオ入力モードの切り替え時(Lib_set_xvideo_input_mode 関数実行時)

に、カメラに対してカメラリンク経由でシリアルコマンドを送信しています。
送信されるコマンドは、各カメラの ini ファイルに記述されています。

- ini ファイルの記述 -

以下の図は、LNX でのカメラリンク対応ボードの ini ファイル（コマンド記述箇所）を図示したものです。
(例 : FVC06 CSB400CL)

ini ファイルには、それぞれコマンドが送られるときに応じてのコマンドを記述しています。

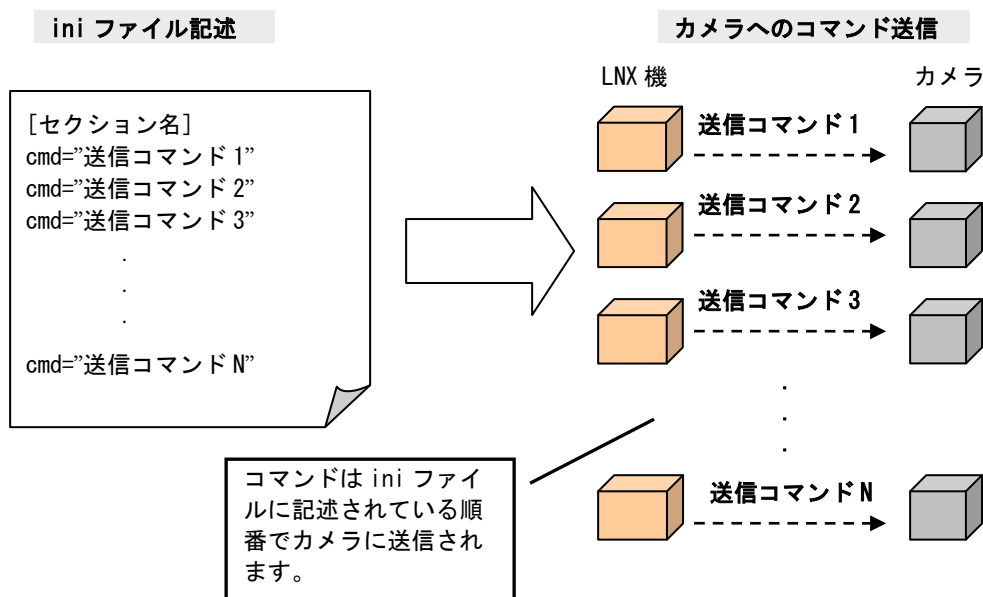
セクション	キー	意味
CmdInitial	cmd	カメラ初期化時のコマンド
CmdNormal	cmd	通常入力モードに変更時のコマンド
CmdTrigger	cmd	トリガを使用した取込モードに変更時のコマンド

各セクションのキー"cmd"の記述仕様は同じです。

cmd="送信コマンド"

"cmd"に記述する送信コマンドには、STX/ETX をのぞいた文字列を記述します。

- コマンドの送信順序 -

各セクションに記述されたコマンドはiniファイルに記述されている順番が、そのままカメラにコマンド送信される順番となります。

- iniファイルの保管場所 -

windows等で、LNX機のCFカードを取り出すと、iniファイルを確認することができます。
FVC06とFHCシリーズでは、CFカード上の保存場所が違うため、ご注意ください。

- FVC06

保存場所: parmsys¥camera¥FVC06_(対応カメラ名).ini

- FHC3312,FHC3313

保存場所: parmsys¥camera¥FHC(3312|3313)_(対応カメラ).ini
parmsys¥camera¥avaldata¥(カメラメーカー名)¥(対応カメラ).ini

FHCシリーズのiniファイルでは、カメラ設定のiniファイルが二種類用意されています。
paramsys¥camera¥avaldata¥(カメラメーカー名)¥以下のiniファイルでは、カメラの取込に関するパラメータが記述されています。
parmsys¥camera以下のファイルには、そのiniファイルの指定パスと、カメラ送信コマンドが記述されています。

3.4　自動送信されるシリアルコマンド一覧

カメラリンク対応ボード(FVC06・FHC3312・FHC3313)使用時には下記のタイミングでカメラに対してシリアルコマンドが自動的に送られます。

・カメラ初期化時

・ビデオ入力モード切替時（通常入力モード/トリガモード）
※ Lib_set_xvideo_input_mode で設定時

送られるコマンドは各カメラの ini ファイルに記載されておりカメラによって異なります。
上記タイミングで送られるコマンドについてカメラごとにまとめました。

自動送信される設定から変更なさりたい場合、アプリケーション内で Lib_capt_send_cmd を実行してご希望の設定コマンドを送信する事が可能です。なお Lib_capt_send_cmd でコマンドを送信し設定を変更しても次回起動時は ini ファイルの情報で再度設定されます。Lib_capt_send_cmd での設定は毎回行うようにしてください。またカメラ ini ファイルを直接変更する方法もあります。ini ファイルの詳細につきましては前述の「3.1.5　FVC06 の ini ファイル仕様」または「3.2.5　FHC3312,FHC3313 の ini ファイル仕様」のページをご覧下さい。

コマンド送信で変更する必要があるケースとしては下記の様な例があります。

〈変更例〉
・TL-7400RCL で露光制御設定をライン周期から:パルス露光に変更したい。

・カメラのゲインを変更したい。

Lib_capt_send_cmd 詳細につきましては該当ライブラリのページをご参照下さい。また送信するコマンドにつきましては、各カメラメーカーの説明書をご覧下さい。

ただ単にコマンドによる設定変更後の映像を確認する目的であれば LNX システムメニューにある「カメラ調整ツール」内のシリアルコンソール(又は同等の機能を持つライブラリ Lib_show_serial_console)をご利用頂くことができます。

**※弊社では標準添付のカメラ ini ファイルを使用してカメラ評価を行っています。
設定するコマンドまたは ini ファイルの直接編集によっては正常に取り込めなくなる可能性もございますので十分にご注意下さい。**

FVC06 対応ラインセンサカメラ

TL-7400RCL　　カメラ ini ファイル = FVC06_TL-7400RCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
ctrl=1	DIP ｽｲｯﾁ/通信設定値有効状態： Com
bit=8	カメラの出力 bit 数： 8bit 出力
expc=0	露光制御設定： ライン周期
[CmdNormal]	通常入力モード移行時のコマンド
expc=0	露光制御設定： ライン周期
sync=0	SYNC 切替： Sync Auto
[CmdTrigger]	トリガモード移行時のコマンド
expc=0	露光制御設定： ライン周期
sync=0	SYNC 切替： Sync Auto※

※このモード(sync=0)は 1 ミリ以上のスキャンレートでは使用できません。
1 ミリ以上のスキャンレートを使用したい場合は sync=2 に設定してください。
エンコーダを使用している場合過露光の現象が出る可能性があります。

NSUF0514　　カメラ ini ファイル = FVC06_NSUF0514.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
H=6	データ転送： 20 MHz で 2 tap 出力
S=0	出力フォーマット： 12 bit 出力データ
[CmdNormal]	通常入力モード移行時のコマンド
M=1	トリガモード： フリーラン＋積算時間設定
[CmdTrigger]	トリガモード移行時のコマンド
M=2	トリガモード： 外部トリガ＋積算時間設定

FHC3312 対応ライセンサ

NSUF0514　　カメラ ini ファイル = FHC3312_NSUF0514CL.ini
NSUF1014　　カメラ ini ファイル = FHC3312_NSUF1014CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
!=5	特殊コマンド：初期時のカメラ設定に復帰
H=9	データ転送： 60 MHz で 1 tap 出力
S=2	出力フォーマット： 8 bit 出力データ
M=1	トリガモード： フリーラン＋積算時間設定
[CmdNormal]	通常入力モード移行時のコマンド
M=1	トリガモード： フリーラン＋積算時間設定
[CmdTrigger]	トリガモード移行時のコマンド
M=2	トリガモード： 外部トリガ＋積算時間設定

NSUFi74　　カメラ ini ファイル = FHC3312_SUFI74.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
x0	駆動 CK：内部 CK(50MHz)
[CmdNormal]	通常入力モード移行時のコマンド
t0	露光モード：フリーラン(プログラマブル設定固定時間)
[CmdTrigger]	トリガモード移行時のコマンド
t1	露光モード：外部トリガ(エッジ周期露光)

FHC3313 対応ライセンサ

PIRANHA3　　カメラ ini ファイル = FHC3313_DALSA_P3_12k.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
rfs	工場出荷時の設定に戻す
[CmdNormal]	通常入力モード移行時のコマンド
sem 7	露光モード設定：カメラ内部同期
[CmdTrigger]	トリガモード移行時のコマンド
sem 3	露光モード設定：カメラ外部同期、トリガ周期による露光

FVC06 対応エリアカメラ

CSB1100CL　カメラ ini ファイル = FVC06_CSB1100CL.ini
CSB4000CL　カメラ ini ファイル = FVC06_CSB4000CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
0x66 0x00	Shutter Mode : 0x00
0x6E 0x01	Update Windowing(0x60 以降のデータをカメラに反映)
[CmdNormal]	通常入力モード移行時のコマンド
0x6A 0x00	Trigger1 :　トリガ極性=負極性　トリガモード OFF
0x6E 0x01	Update Windowing(0x60 以降のデータをカメラに反映)
[CmdTrigger]	トリガモード移行時のコマンド
0x6A 0x01	Trigger1 :　トリガ極性=負極性　トリガモード ON
0x6B 0x01	Trigger2 :　露光モード=Pulse_Mode(※)
0x6E 0x01	Update Windowing(0x60 以降のデータをカメラに反映)

※シャッタスピードがトリガパルス幅に依存するモード

CS3970CL　カメラ ini ファイル = FVC06_CS3970CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
0x70 0x00	SubSampling Y　OFF
0x81 0x00	Window Enable　OFF
0x90 0x00	Output Bit Information　Bit Depth
0x6E 0x01	Update Windowing
[CmdNormal]	通常入力モード移行時のコマンド
0x6A 0x00	Trigger1 :　トリガ極性=負極性　トリガモード OFF
0x6E 0x01	Update Windowing
[CmdTrigger]	トリガモード移行時のコマンド
0x6A 0x01	Trigger1 :　トリガ極性=負極性　トリガモード ON
0x6B 0x01	Trigger2 :　露光モード=Pulse
0x6E 0x01	Update Windowing

CV-M2CL　カメラ ini ファイル = FVC06_CV-M2CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
EB=0	Echo Back = OFF
OS=1	Output select = dual chan.
BA=0	Output bit allocation = 10bit
SC=0	Scanning format = full frame
VB=0	Vertical Binning = OFF
[CmdNormal]	通常入力モード移行時のコマンド
TR=0	Trigger mode = normal
[CmdTrigger]	トリガモード移行時のコマンド
TR=3	Trigger mode = pulse width
TP=0	Trigger polarity = active low
LS=0	LVAL synchronous accum = syn. accum

VCC-G22V31CL　　カメラ ini ファイル = FVC06_VCC-G22V31CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
000000W005000	SCAN MODE = NOMAL SCAN MODE
000000W013000	8bit OUTPUT DATA
[CmdNormal]	通常入力モード移行時のコマンド
000000W004000	TRIGGER MODE = NOMAL TRIGGER MODE (TRIGEER MODE OFF)
[CmdTrigger]	トリガモード移行時のコマンド
000000W004002	TRIGGER MODE = PULSE WIDTH TRIGGER MODE(※)
000000W011000	TRIGGER POLARITY = POSITIVE INPUT
000000W012000	INPUT TRIGGER PORT SELECT = 26pin Camera Link
000000W014000	Trigger Reset Type = V-SYNC Reset

※トリガのパルス幅でシャッタスピードを設定します。

KP-F200CL-S1　　カメラ ini ファイル = FVC06_KP-F200CL-S1.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = 1/30
01FF0114000000	DATA BIT = 8bit
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	MODE = NORMAL
[CmdTrigger]	トリガモード移行時のコマンド
01FF0110000000	TRIGGER = NORMAL
01FF0104010000	MODE = 1TRIGGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE

KP-F31SCL　　カメラ ini ファイル = FVC06_KP-F31SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
01FF0113000000	VERTICAL 2 PIXEL ADDITION = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE

KP-F80SCL　　カメラ ini ファイル = FVC06_KP-F80SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE
01FF0102010000	TRIGGER HD RESET = RESET

KP-F200SCL　　カメラ ini ファイル = FVC06_KP-F200SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE

KP-F230SCL　　カメラ ini ファイル = FVC06_KP-F200SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
01FF0113000000	VERTICAL 2 PIXEL ADDITION = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE
01FF0102010000	TRIGGER HD RESET = RESET

KP-F500SCL　　カメラ ini ファイル = FVC06_KP-F500SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0112000000	CONFIGURATION BASE
01FF0114000000	DATA BIT = 8bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
01FF0113000000	VERTICAL 2 PIXEL ADDITION = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIG
01FF010F000000	TRIGGER POLARITY = POSITIVE
01FF0102010000	TRIGGER HD RESET = RESET

FHC3312 対応エリアカメラ

BM-500CL　　カメラ ini ファイル = FHC3312_BM-500CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
EB=0	すべての通信をエミュレータスクリーン上で表示しない
BA=0	ビットアロケーション = 8bit 出力
SC=0	部分読み出し
VB=0	垂直ビニング
[CmdNormal]	通常入力モード移行時のコマンド
TR=0	Trigger Mode = Normal (Continuous)
[CmdTrigger]	トリガモード移行時のコマンド
TR=2	Trigger Mode = 2=PWC(Pulse width control)
TP=0	Trigger Polarity = Active Low
LS=0	

VCC-G60FV11CL　　カメラ ini ファイル = FHC3312_VCC-G60FV11CL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
000000W013000	Output Data Select = 8bit Output Data
[CmdNormal]	通常入力モード移行時のコマンド
000000W004000	Trigger Mode = Normal Mode (Trigger Mode OFF)
[CmdTrigger]	トリガモード移行時のコマンド
000000W004002	Trigger Mode = Pulse Width Trigger Mode(※)
000000W011000	Trigger Polarity = Positive Input
000000W012000	Trigger Input Select = カメラリンク対応コネクタ
000000W015000	Partial Scan Mode = Full Frame Scan Mode

※トリガのパルス幅でシャッタスピードを設定します。

CSCU30BC18　　カメラ ini ファイル = FHC3312_CSCU30BC18.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
9000	Output bit = 12bits
6A00	Restart Reset = Off　Polarity = NEG　RANDOM = OFF
[CmdNormal]	通常入力モード移行時のコマンド
6A00	Restart Reset = Off　Polarity = NEG　RANDOM =OFF
6E01	Update
[CmdTrigger]	トリガモード移行時のコマンド
6A01	Restart Reset = Off　Polarity = NEG　RANDOM = ON
6B01	RandomMode = PLS
6E01	Update

CSC12M25BMP19　　カメラ ini ファイル = FHC3312_CSC12M25BMP19.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
91,0	シャッターモード　ノーマルシャッタ OFF
[CmdNormal]	通常入力モード移行時のコマンド
91,0	シャッターモード　ノーマルシャッタ OFF
[CmdTrigger]	トリガモード移行時のコマンド
91,2	シャッターモード　ランダムトリガシャッタ
92,1	ランダムトリガモード　パルスモード

KP-F31SCL　　カメラ ini ファイル = FHC3312_KP-F31SCL.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
01FF0113000000	VERTICAL 2 PIXEL ADDITION = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE

KP-F230SCL　　カメラ ini ファイル = FHC3312_KP-F230SCL.ini

コマンド	
[CmdInitial]	カメラ初期化時のコマンド
01FF0108000000	SHUTTER SPEED = OFF
01FF0114010000	DATA BIT = 10bit
01FF0115010000	VD/FVAL = FVAL
01FF0116010000	HD/LVAL = LVAL
01FF011E000000	PARTIAL SCAN MODE = OFF
01FF0113000000	VERTICAL 2 PIXEL ADDITION = OFF
[CmdNormal]	通常入力モード移行時のコマンド
01FF0104000000	TRIGGER MODE = OFF
[CmdTrigger]	トリガモード移行時のコマンド
01FF0104020000	TRIGGER MODE = 1TRIGER
01FF010F000000	POLARITY(TRIG-A) = POSITIVE
01FF0102010000	TRIGGER HD RESET = RESET

FHC3313 対応エリアカメラ

CSC12M25BMP19　　カメラ ini ファイル = FHC3312_CSC12M25BMP19.ini

コマンド	説明
[CmdInitial]	カメラ初期化時のコマンド
87,8	出力ビット数　8bit
91,0	シャッターモード　ノーマルシャッタ OFF
[CmdNormal]	通常入力モード移行時のコマンド
91,0	シャッターモード　ノーマルシャッタ OFF
[CmdTrigger]	トリガモード移行時のコマンド
91,2	シャッターモード　ランダムトリガシャッタ
92,1	ランダムトリガモード　パルスモード

付録. 各ライブラリの処理速度一覧

○計測対象機種

901	CSC901NT
902	FV902　model-3（Celeron　433MHz）
903	CSC903
904	FV904
FVL/LNX	FV2000

各ライブラリの実行所要時間欄の

空　白 ━━► 使用不可

――――― ━━► 条件により変化

・・・・・・ ━━► 測定不能（所要時間が0.001ミリ秒未満）

を表しています。

基本ライブラリ実行所要時間代表値一覧

単位　ミリ秒

項番	ライブラリ名	所要時間					備　考
		901	902	903	904	FVL/LNX	
1	Lib_2ch_indication	136/213	103/105	167/167	161/187	103/86	2値/グレイ
2	Lib_camera_chain_freeze	152.994	70.801	133.463	82.194		グレイ　2フレーム
3	Lib_capt_get_frame_index					……	
4	Lib_capt_grab_end					———	
5	Lib_capt_grab_start					———	
6	Lib_capt_wait_event					———	
7	Lib_channel	0.163	0.001	0.109	0.081	0.001	NO_WAIT
8	Lib_ctrl_strobo	———			———		
9	Lib_dis_ext_trg			———			
10	Lib_display_control	0.109	0.001	0.054	0.054	0.001	
11	Lib_ena_ext_trg			———			
12	Lib_ex_disp_bin_memory	———	———				
13	Lib_ex_disp_gray_memory	———	———				
14	Lib_ex_video_transmit	———	———				
15	Lib_extraction_display_memory	———	———	———	———		
16	Lib_frame_chain_freeze	121.256	75.683	100.097	82.194	82.000	グレイ　2フレーム
17	Lib_framework	7.324	1.628	2.368	0.814	47.000	(0,0)-(511,479)
18	Lib_freerun	0.109	……	0.027	0.027	31.000	
19	Lib_freeze	———	———	———	———	———	
20	Lib_freeze_tm9700	10+Lib_freeze時間			10+Lib_freeze時間		
21	Lib_get_exposure	———	……	0.278		……	

基本ライブラリ実行所要時間代表値一覧

単位　ミリ秒

項番	ライブラリ名	所要時間					備　　考
		901	902	903	904	FVL/LNX	
22	Lib_get_frame_ratio	0.027	……	……	……	……	
23	Lib_get_last_field_id	0.027	……		……		
24	Lib_get_video_clock_rate		……	……			
25	Lib_get_video_input_mode	0.027	……	……	……	……	
26	Lib_get_video_status		———			———	
27	Lib_get_xvideo_input_mode		……			……	
28	Lib_GetCaptureType		……			……	
29	Lib_gray_scale	0.027	……	……	……	……	
30	Lib_input_video_control	0.136	0.001	0.054	0.054	……	
31	Lib_random_bin_level			0.278			
32	Lib_random_video_channel	219.73/ 209.15/ 201.00/ 196.94			44.759		1/60, 1/100 1/250, 1/500～
33	Lib_set_both_mode			0.015			
34	Lib_set_exposure		0.002	0.278		0.001	
35	Lib_set_field_squence	43.945	……		50.049		
36	Lib_set_hline		……			……	
37	Lib_set_sim_bin_memory		……			……	
38	Lib_set_sim_channel		……			……	
39	Lib_set_sim_gray_memory		……			……	
40	Lib_set_stage_input	……	……	31.169	……	……	

基本ライブラリ実行所要時間代表値一覧

単位　ミリ秒

項番	ライブラリ名	所要時間					備　考
		901	902	903	904	FVL/LNX	
41	Lib_set_strobo	0.02	・・・・・・		———		
42	Lib_set_video_clock_rate		・・・・・・	・・・・・・			
43	Lib_set_video_input_mode	29.297	0.032	1.112	29.297	245.000	
44	Lib_set_vline		・・・・・・	0.036	———	・・・・・・	
45	Lib_set_xvideo_input_mode		———			245.000	
46	Lib_stage_input	———	11.393	———	8.952	・・・・・・	
47	Lib_stop_freerun					133.500	平均値
48	Lib_strobo_freeze	———	———	———	———		
49	Lib_video_bin_level	0.054	0.000	0.027	0.027	・・・・・・	
50	Lib_video_channel	0.190	0.001	0.136	0.081	・・・・・・	
51	Lib_video_offset	0.054	・・・・・・	0.027	0.027	・・・・・・	
52	Lib_video_transmit	6.5/67.5	13.0/6.5	17/24.4	4.8/11.3	17/15	2値/グレイ
53	Lib_window	22.786	11.393	30.762	8.952	15.000	(0,0)-(511,479)
54	Lib_xcamera_chain_freeze	113.932	79.752	133.463	99.609		(濃淡　2フレーム)
55	Lib_xframe_chain_freeze	113.932	79.752	100.097	83.008	92.000	(濃淡　2フレーム)
56	Lib_xfreeze		・・・・・・			・・・・・・	
57	Lib_xvideo_channel	———	———	———	———	———	WAITING
58	Lib_xvideo_transmit	7.3/67.5	13.0/6.5	16.9/24.3	4.8/11.3	18/17	2値/グレイ

Lib_freeze 実行所要時間代表値

単位　　ミリ秒

計測対象機種	基本時間	所要時間（基本時間に加算されます）			
		(Lib_input_video_control)の設定		(Lib_display_control)の指定	
		画像メモリ取込		画像転送(TRANSMIT 指定)	
		グレイ	2値	グレイ	2値
901NT	33～50	0	0	0	0
FV902model3 RICE, PLUM	33～50	0	4	9	21
902ST-CATS	0～33	24	4	26	39
903	33～66	0	4	6	51
904	33～66	0	0	12	5
FVL/LNX	33～66	0	0	15	17

・上記所要時間はビデオ入力モードが“通常入力モード”の場合です。
・902ST-CATS の基本時間は、ほとんどの場合 0～16 ミリ秒です。
　連続して Lib_freeze を実行する場合やチャネル切替え直後に基本時間が 0～33 になる場合があります。

索　　引

2

C

D

E

F

G

I

索　　引